



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



DERİN ÖĞRENME YÖNTEMLERİ İLE GÖRÜNTÜ İŞLEME UYGULAMALARI

ABDULSAMET AKTAŞ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN

Dr.Öğr Üyesi Önder DEMİR

EŞ-DANIŞMAN

Dr.Öğr Üyesi Buket DOĞAN

İSTANBUL, 2020



MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



DERİN ÖĞRENME YÖNTEMLERİ İLE GÖRÜNTÜ İŞLEME UYGULAMALARI

ABDULSAMET AKTAŞ

523616020

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Yüksek Lisans Programı

DANIŞMAN

Dr.Öğr Üyesi Önder DEMİR

EŞ-DANIŞMAN

Dr.Öğr Üyesi Buket DOĞAN

İSTANBUL, 2020

ÖNSÖZ

Tez çalışmam süresince desteklerini, rehberliklerini, zamanlarını, yardımlarını ve sabırlarını esirgemeyen değerli hocalarım Dr. Öğr. Üyesi Buket Doğan'a ve Dr. Öğr. Üyesi Önder Demir'e,

Bana olan inançları ve koşulsuz sevgileri ile hayatımın her anında yanımda olan çok kıymetli Annem Ayla Aktaş ve Babam İbrahim Aktaş'a,

Destekleri, saygı ve sevgileri ile her an yanımdaymış hissi veren çok değerli kardeşlerim, ablam ve ailesine,

Yardımları ve destekleriyle yanımda olan değerli çalışma arkadaşlarıma,

Yardımlarından dolayı kısmi zamanlı olarak yanımda çalışan sevgili öğrencilerime,

Tez çalışmam süresince kullandığım Derin Öğrenme Bilgisayarını temin eden Marmara Üniversitesi Marmara Bilimsel Araştırma Projeleri birimine,

Teşekkürü bir borç bilirim.

Ocak, 2020

Abdulsamet AKTAŞ

İÇİNDEKİLER

	SAYFA
ÖNSÖZ	i
İÇİNDEKİLER	ii
ÖZET	iv
ABSTRACT	v
SEMBOLLER	vi
KISALTMALAR	vii
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	x
1. GİRİŞ	1
1.1. Yapay Zeka	2
1.1.1. Derin Öğrenme	4
1.1.2. Evrişimsel Sinir Ağları (ESA)	6
1.1.3. You Only Looked Once (YOLO)	7
1.2. Dokunsal Parke Yüzeyleri	7
1.3. Amaç ve Önem	10
1.4. Literatür Araştırması	11
1.4.1. Görüntü İşleme İle İlgili Çalışmalar	11
1.4.2. Dokunsal Parke Yüzeyi Tespiti İle İlgili Çalışmalar	15
2. MATERYAL VE YÖNTEM	19
2.1. Kullanılan Veri Seti	19

2.1.1.	Veri Çoğullama	20
2.1.2.	Veri Temizleme	21
2.1.3.	Veri Etiketleme	21
2.1.4.	Veri Setinin Normalize Edilmesi	22
2.1.5.	Test ve Eğitim Sınıflarının Belirlenmesi	24
2.2.	Kullanılan Yöntemler	24
2.2.1.	YOLO(You Only Look Once)	24
2.2.2.	YoloV3	27
2.2.3.	Yoğun Bağlantılı Sinir Ağları (DenseNet)	31
2.2.4.	Önerilen Algoritma YOLO-V3 Dense	32
3.	BULGULAR VE TARTIŞMA	35
3.1.	Kullanılan Metrik Yöntemler	35
3.1.1.	Kesinlik (Precision), Duyarlılık (Recall) , F1 skor ve mAP	35
3.1.2.	Kayıp Fonksiyonu (Loss Function)	37
3.2.	Modellerin Karşılaştırılması	38
4.	SONUÇLAR	45
	KAYNAKLAR	46
	ÖZGEÇMİŞ	51

ÖZET

DERİN ÖĞRENME YÖNTEMLERİ İLE GÖRÜNTÜ İŞLEME UYGULAMALARI

Gerçek zamanlı çalışan sistemlerde görüntü işleme uygulamaları yapmak son zamanlarda oldukça popüler olan bir konu haline gelmiştir. Yapay zekâ alanının alt dallarından biri olan derin öğrenme yöntemleri ve görüntülerden nesne tespiti yapma alanında kullanılan görüntü işleme algoritmaları birlikte kullanılarak, otonom otomobiller, otonom insansız hava araçları, yardımcı robot teknolojileri, engelli ve yaşlı bireyler için asistan teknolojileri gibi birçok alanda uygulamalar geliştirilmektedir. Gerçekleştirilen bu çalışmada görme engelli bireyler için yardımcı bir teknoloji olarak derin öğrenme yöntemleri kullanılarak dokunsal parke yüzeylerinin tespit edilmesi amaçlanmıştır.

Geleneksel görüntü işleme algoritmalarının aksine bu çalışmada derin öğrenme yöntemleri ile görüntü işleme algoritmaları birlikte kullanılmıştır. Nesne tespit etme alanında en iyi modellerden biri olan YOLO-V3 modeli DenseNet modeli ile birleştirilerek YOLOV3-Dense modeli oluşturulmuştur. YOLO-V2, YOLO-V3 ve YOLOV3Dense modelleri tarafımızca oluşturulmuş olan ve içerisinde 4580 etiketli görsel bulunan Marmara Dokunsal Parke Yüzeyi(MDPY) veri seti üzerinde ayrı ayrı eğitildikten sonra performansları test veri seti üzerinde birbirleri ile karşılaştırılmıştır. %89 F1-skor, %92 ortalama hassasiyet ve %81 IoU değerleri ile YOLOV3-Dense modelinin dokunsal parke yüzeyi tespit etmede diğer modellerden daha iyi olduğu gözlemlenmiştir. Saniyede 60 kare çalışma hızı ile YOLOV3-Dense modeli gerçek zamanlı çalışan sistemlerde de kullanılabilir.

Ocak, 2020

Abdulsamet AKTAŞ

ABSTRACT

IMAGE PROCESSING APPLICATIONS WITH DEEP LEARNING METHODS

Image processing applications in real-time systems have become a popular topic in recent times. Deep learning methods, which are one of the sub-branches of artificial intelligence, and image processing algorithms are used together to develop applications in many fields such as autonomous automobiles, autonomous unmanned aerial vehicles, assistive robot technologies, assistant technologies for disabled and elderly individuals. This study aims to detect the tactile paving surfaces with deep learning methods in order to design an assistive technology system that can be used by visually impaired individuals, autonomous vehicles and robots.

Contrary to traditional image processing algorithms, deep learning methods and image processing algorithms are used together in this study. The YOLO-V3 model, which is one of the best methods of object detection, is combined with the DenseNet model to create the YOLOV3-Dense model. YOLO-V2, YOLO-V3 and YOLOV3Dense models were trained on the Marmara Tactile Paving Surface (MDPY) dataset, which was created by the researchers and included 4580 images, and their performances were compared with each other on the test dataset. It was observed that YOLOV3-Dense model is better than other models in detecting tactile paving surface with 89% F1-score, 92% mean average Precision(mAP) and 81% IoU values. At 60 frames per second, the YOLOV3-Dense model can also be used in real-time systems.

January, 2020

Abdulsamet AKTAŞ

SEMBOLLER

X : Etiketli verinin orta noktasına ait X koordinatı.

Y : Etiketli verinin orta noktasına ait Y koordinatı.

W : Etiketli verinin genişliği

H : Etiketli verinin yüksekliği

dW : Görüntü genişliği

dH : Görüntü yüksekliği

w_{ij} : Nöron ağırlık değeri

μ : Ortalama

λ : Sabit Katsayı

Σ : Toplam

I^{obj} : Nesne bulunma durum

I^{noobj} : Nesne bulunmama durumu

P : Olasılık

KISALTMALAR

Cnn	: Convolutional Neural Networks
Dpy	: Dokunsal Parke Yüzeyi
Esa	: Evrişimsel Sinir Ağları
Iou	: Intersection Over Union
Fp	: Yanlış-Pozitif(False-Positive)
Fn	: Yanlış-Negatif(False-Negative)
Map	: Ortalama Hassasiyet (Mean Average Precision)
Mdpy	: Marmara Dokunsal Parke Yüzeyi
Ssd	: Single Shot Detector
Tn	: Doğru-Negatif(True-Negative)
Tp	: Doğru-Pozitif(True-Positive)
Yolo	: You Only Look Once

ŞEKİL LİSTESİ

SAYFA

Şekil 1.1 Yapay Zeka Tarihsel Gelişim Süreci.....	4
Şekil 1.2. Yapay Sinir Ağı Yapısı.[18].....	5
Şekil 1.3. Nöron Matematiksel Modeli.[19].....	5
Şekil 1.4. LeNET-5 Evrişimsel Sinir Ağı Mimarisi.[21]	6
Şekil 1.5. Dokunsal Parke Yüzeyleri.[26]	8
Şekil 1.6. Farklı Ülkelere Göre Dokunsal Parke Yüzeyleri.[26].....	8
Şekil.1.7. Yanlış Kurulum Yapılan Dokunsal Parke Yüzeyleri.[26].....	9
Şekil 2.1 Sistem Mimarisi.	19
Şekil 2.2.İç Mekânlara Ait Veri Setinde Bulunan Örnek Görüntüler.	19
Şekil 2.3. Dış Mekânlara Ait Veri Setinde Bulunan Örnek Görüntüler.	20
Şekil 2.4. Veri Çoğullama Örnek Görüntüler.....	20
Şekil 2.5. (a) Veri setinde bulunan örnek görüntü. (b) Çoğullama işleminden sonra elde edilen örnek görüntü.	21
Şekil 2.6. LabelTool Veri Etiketleme Programı[54]	22
Şekil 2.7. Etiketlenen Verinin Koordinatları.[23]	23
Şekil 2.8. Çoklu görev yapabilen bir insan.....	24
Şekil 2.9. Hızlı R-CNN Mimarisi.[48]	26
Şekil 2.10. YOLO Tespit Şeması.	26
Şekil 2.11. Darknet-53 mimarisi.[23].....	29
Şekil 2.12. YOLO-V3 Ağ Yapısı.[57].....	30
Şekil 2.13. COCO veri seti üzerinde 0.5 IoU değeri ile nesne tespit yöntemlerine ait mAP değerlerinin ve tespit sürelerinin karşılaştırılması.[58]	31
Şekil 2.14. Yoğun Bağlantılı Sinir Ağları.[59].....	32
Şekil 2.15. YOLOV3-Dense Ağ parametreleri.	33
Şekil 2.16. YOLOV3-Dense YSA mimarisi.	34
Şekil 3.1. Modellere ait yitim değerleri.....	39
Şekil 3.2. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,25 ..	39
Şekil 3.3. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,50 ..	40

Şekil 3.4. DPY Tespit Edilmiş Örnek Görüntüler.	41
Şekil 3.5. Doğrulama veri setine ait örnek görüntüler.....	42

TABLO LİSTESİ

	SAYFA
Tablo 1.1. Dokunsal Parke Yüzeyi Tespiti İle İlgili Literatürde Bulunan Çalışmalar ...	18
Tablo 3.1. Karmaşıklık Matrisi.	36
Tablo 3.2. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri.....	40
Tablo 3.3. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri.....	40
Tablo 3.4. Örnek veri setine ait güven skoru değerleri.	42
Tablo 3.5. Diğer yöntemler ile karşılaştırma tablosu.	43

1. GİRİŞ

Dünya genelinde yaklaşık olarak 1.3 milyar insan görme bozukluğu yaşamaktadır. Bu insanların yaklaşık 36 milyonu görme yetisini tamamen kaybetmiştir[1]. Görme engeline sahip bireyler daha rahat hareket edebilmek için rehber köpekler ve baston gibi yardımcı unsurlar kullanmaktadırlar. Bu yardımcı unsurların yanı sıra gerek dış mekânlarda gerekse iç mekânlarda daha rahat hareket edebilmelerini sağlamak amacıyla zemine döşenmiş kabartmalı yüzeyler bulunmaktadır. Bu yüzeylerin şekilleri, renkleri ve isimleri ülkelere göre değişebilmektedir [2].

Yaygın olarak dokunsal parke yüzeyi (DPY)olarak adlandırılan ve sarı renkte olan bu yüzeyler görme engelli bireylerin hareket etmelerinde kolaylık sağlasalar da kesin bir çözüm olamamışlardır. Zamanla bu yüzeylerin eskimesi, yüzeylerin olması gerektiği gibi konumlandırılmaması, hareketi engelleyecek şekilde yüzeylerin üzerinde veya etrafında engellerin olması gibi problemler yaşanabilmektedir. Bu problemleri görme engelli kişilerin fark edebilmesi ve problemi çözebilmesi her zaman mümkün olmamaktadır. Problemleri çözmek amacıyla günümüz teknolojileri kullanılarak görme engelli bireylere yardımcı olabilecek uygulamalar geliştirilmektedir. Görme engelli bireylerin yanı sıra otonom bir şekilde hareket edebilen robotlar ve araçlar da bu yardımcı uygulamalara ihtiyaç duymaktadır. Geliştirilen yardımcı uygulamalarda görüntü işleme algoritmaları [3-5], ultrasonik sensörler [6] ve ışık kodlama teknolojisi [7]gibi yöntemler kullanılmıştır. Görüntü işleme algoritmaları ile DPY'nin renk ve şekil öznitelikleri kullanılarak, ultrasonik sensörler ve ışık kodlama teknolojisi ile yardımcı elemanların birbiri ile haberleşmesi ve bilgi aktarımı sağlanarak DPY tespiti yapılmıştır.

Son yıllarda yapılan bilgi depolama teknolojilerindeki gelişmelerin etkisiyle işlenmemiş ham veri miktarı oldukça artmıştır [8]. Bu artış ile birlikte paralel hesaplama yöntemleri ve yüksek işlem gücü, makine öğrenmesi ve görüntü işleme gibi alanlar için vazgeçilmez bir hale gelmiştir. Bütün bu gelişmeler ile birlikte işlem gücü yüksek ekran

kartları üretilebilir hale gelmiştir. Bu gelişim sayesinde ekran kartları üzerinde paralel hesaplama imkânı veren yeni kütüphaneler oluşturulmuştur. Oluşturulan kütüphaneler ile ekran kartları üzerinde aritmetik işlemler yapmak oldukça kolay ve kullanışlı bir hale gelmiştir. Bu gelişim, çok fazla maliyet, zaman ve yüksek işlem gücü gerektiren hesaplama işlemlerini daha kısa sürede ve kullanışlı bir şekilde yapmayı mümkün kılmıştır. Oluşturulan bu hesaplama ve işlem gücü, yapay sinir ağlarında derin ağ olarak adlandırılan çok katmanlı ağlar ile çalışmayı mümkün kılmıştır. Oluşturulan bu derin ağlar sayesinde görüntü işleme alanında oldukça büyük ilerlemeler kaydedilmiştir. Çok katmanlı sinir ağları görüntülerde bulunan nesnelerin sınıflandırılması ve tespiti ile ilgili yapılan yarışmalarda normal makine öğrenmesi yöntemlerinden çok daha iyi dereceler alarak alanının en iyisi olduğunu kanıtlamıştır[9]. Alex ve arkadaşları [10] tarafından 2012 ImageNet [11] sınıflandırma yarışmasında çok katmanlı evrimsel sinir ağları kullanmış ve %16.42 hata oranı ile derin öğrenme yöntemlerinin diğer sığ ağlara göre daha iyi bir performans gösterdiği kanıtlanmıştır. Derin öğrenme yöntemleri kullanılarak oluşturulan ağlardan alınan başarı giderek artmıştır. ImageNet verilerini sınıflandırmada en iyi insan hata seviyesi %5.1'dir [12]. PReLU-Net %4,94 hata oranına ulaşarak insan doğruluk seviyesini aşan ilk yapay sinir ağı haline gelmiştir[13]. Bu değer ilerleyen yıllarda daha küçülerek 2019 yılında EfficientNet [14] ile %2.9 olmuştur.

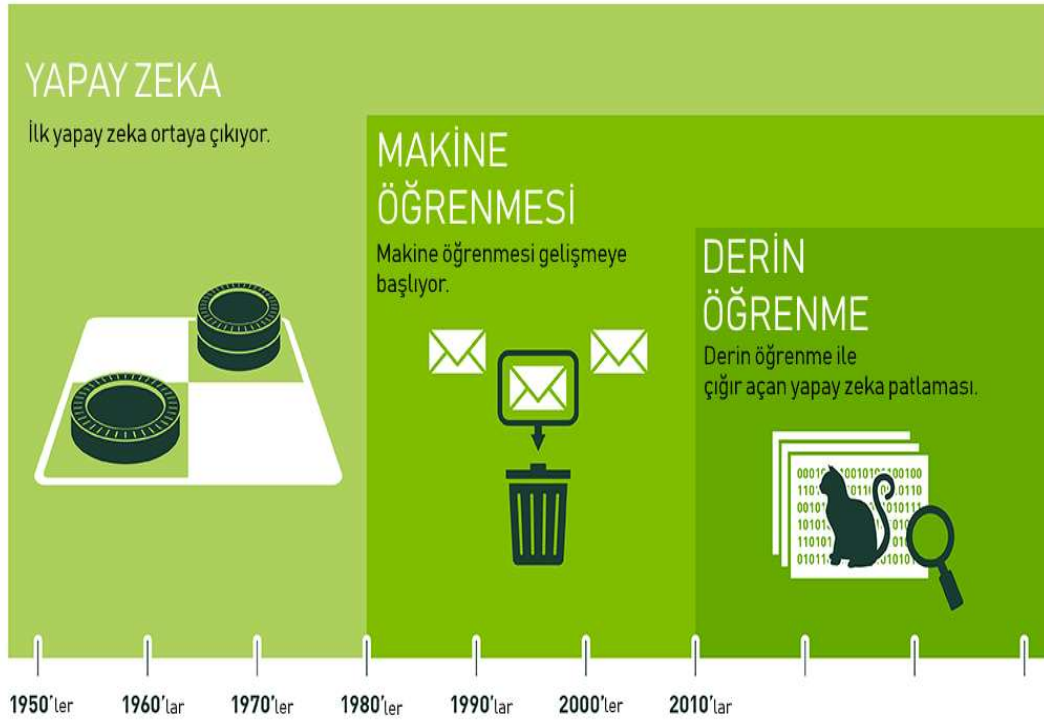
1.1. Yapay Zeka

Yapay zekâ kavramı resmi olarak ilk defa 1955 yılında Honever, New Hampshire Dartmouth College'da yapılan bir konferansta John McCarthy tarafından kullanılmıştır[15]. Yapay zekâ, kendi kendine öğrenebilme, muhakeme yapabilme ve mantıklı kararlar verme gibi insana ait becerileri makinelere uyarlamayı amaçlayan bir bilim alanıdır. Yapay zekânın tarihsel gelişimine bakıldığında ise 1970 yılının bir dönüm noktası olduğu görülmektedir. Bu tarihten önce birçok araştırmanın yapıldığı ve 1969 yılında XOR probleminin çözülememesi nedeni ile araştırmaların durduğu görülmektedir. 1970 yılından sonra sınırlı sayıda araştırmacının çalışmalarını sürdürmeleri ve XOR problemini çözmeleri sonucunda yapay sinir ağlarına olan ilgi

yeniden alevlenmiştir. Bu çalışmalar hem yapay zekâ hem de donanım teknolojisindeki gelişmeler ile de desteklenmiştir. Artık bilgisayarların da öğrenebileceğini herkes kabul etmekte ve bu teknolojiye faydalanmak istemektedir[16]. 1970 yılından sonraki 10 yıl içerisinde birbirinden farklı yeni modeller geliştirilmiştir. Bu modellerden biri olan makine öğrenmesi, basit bir şekilde verilerin ayrıştırılması için bir algoritma kullanma, ayrıştırılan verileri öğrenme ve daha sonra yeni gelen verilerin ne olduğuna dair tahmin yürütme işlemlerini yapmak gibi becerileri olan eğitilmiş bir makinedir. Makine öğrenmesi ile farklı bilgi işlem teknikleri ortaya çıkmıştır. Bu tekniklerin en önemlilerinden biri insan beyininde bulunan sinir ağları temel alınarak geliştirilmiş olan yapay sinir ağlarıdır. İnsan beyininde bulunan sinir ağının çok daha az karmaşık yapıda ve sığ katmanlı bir şekilde bilgisayar ortamında gerçekleştirilmiş halidir. Yapay sinir ağlarının sığ katmanları yerine çok daha fazla katmanlı bir şekilde modellenmesi mümkündür. Bilgisayar grafik kartlarındaki gelişmeler, bu kartların programlanabilmesi ve paralel bir şekilde çalışabilmesi işlemcilerin üzerindeki yükü ortadan kaldırarak çok katmanlı yapay sinir ağları ile çalışabilmeyi mümkün kılmıştır. Çok katmanlı ve karmaşık sistemler tasarlanarak grafik kartları üzerinde paralel bir şekilde hesaplama yapabilen ve çok daha kısa sürede işlemleri tamamlayabilen yeni yapay zekâ modelleri tasarlanmıştır. Tasarlanan bu yeni modele makine öğrenmesinin bir alt dalı olan derin öğrenme ismi verildi. Derin öğrenme, verilerin özniteliklerini ve özelliklerini öğrenmek için çoklu işlem katmanlarından oluşan hesaba dayalı modeller oluşturabilme imkânı sağlamaktadır. Bu modeller konuşma tanıma, görsel nesne tanıma, nesne algılama ve ilaç keşfi ve genom bilimi gibi diğer birçok alanda yapılan çalışmalara büyük katkıda bulunmuşlardır. Derin evrişimli ağlar görüntü, video, konuşma ve ses işlemede büyük gelişmeler yapılmasına olanak sağlarken, tekrarlayan ağlar metin ve konuşma gibi sıralı verilerde yaygın bir şekilde kullanılmaktadır[17]. Şekil 1.1’de yapay zekâ tarihsel gelişim sürecini görmek mümkündür.

1.1.1. Derin Öğrenme

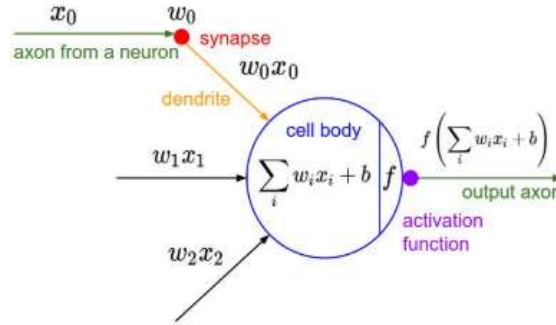
Derin öğrenme, yapay sinir ağları algoritmalarını kullanarak çok katmanlı mimarilerde çok boyutlu veriler ile çalışma imkânı sağlayan, makine öğrenmesi alanının bir alt dalıdır. Derin öğrenme metotları sayesinde doğal dil işleme, görüntü işleme, görsel nesne tespiti, ilaç keşfi, vb. alanlarda ciddi bir şekilde başarı oranı artmıştır [17]. Derin öğrenme, geri yayılım algoritmasını kullanarak çok boyutlu veri setlerinin karmaşık yapısını keşfeder. Bu işlemi, her katmanda hesaplanan parametrelerin değerlerini bir önceki katmanda elde edilen değerler ile karşılaştırarak nasıl bir değişiklik yapılması gerektiğine karar vererek yapar. Derin öğrenmenin temelinde çok katmanlı yapay sinir ağları bulunmaktadır. Basit bir yapay sinir ağı modeli Şekil 1.3'te görmek mümkündür[18]. Şekil 1.3'te bulunan



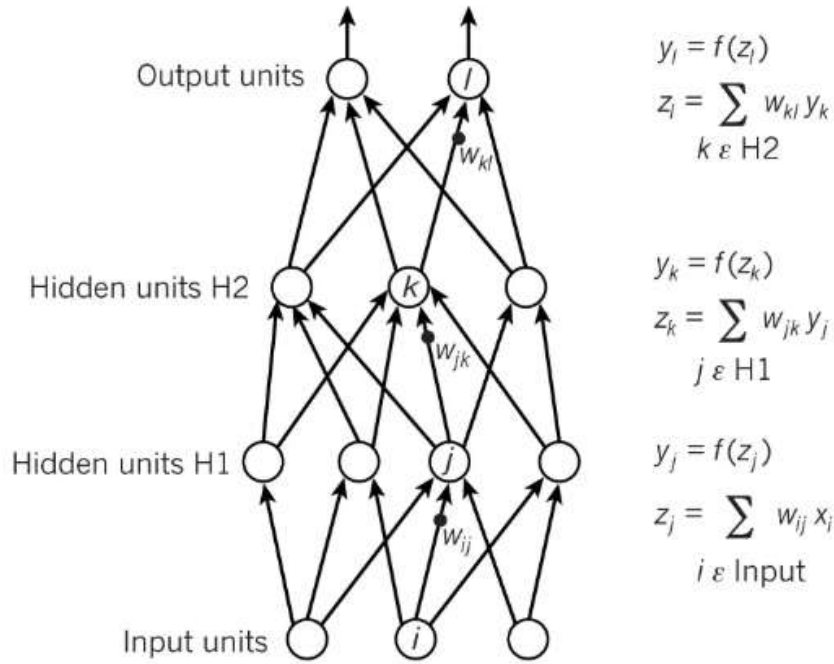
Şekil 1.1 Yapay Zeka Tarihsel Gelişim Süreci.

nöronlar arasındaki her bir bağlantı w_{ij}, w_{jk}, w_{kl} ağırlıkları ile ilişkilidir. Şekil 1.2’de sinir ağında bulunan bir nöronun matematiksel modelini görmek mümkündür[19].

Derin öğrenme metotları danışmanlı öğrenme ve danışmansız öğrenme olarak iki ana gruba ayrılmaktadır [20]. Danışmansız öğrenmede veri setinde veri ile alakalı herhangi bir etiket mevcut değildir. Öğrenme işlemi etiketsiz veri ile yapılmaktadır. Danışmansız öğrenmenin aksine danışmanlı öğrenmede veriler istenilen sonuca göre uygun bir şekilde



Şekil 1.3. Nöron Matematiksel Modeli.[19]

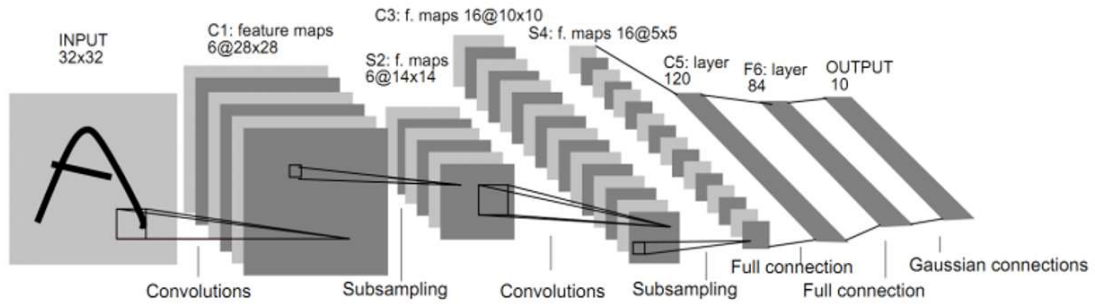


Şekil 1.2. Yapay Sinir Ağı Yapısı.[18]

etiketlenmekte ve sisteme etiketli bir şekilde verilmektedir. Yapılan çalışmada derin öğrenme yöntemlerinden danışmanlı öğrenme metodu kullanılmıştır. Araştırmacılar tarafından toplanan veriler etiketlenerek eğitim ve test işlemleri yapılmıştır.

1.1.2. Evrişimsel Sinir Ağları (ESA)

Evrişimsel Sinir Ağları(Convolutional Neural Network) öğrenebilir ağırlık değerlerine sahip sıradan yapay sinir ağları ile oldukça benzerdir. Sinir ağında bulunan her bir nöron verileri girdi olarak alır, verileri işler ve doğrusal olmayan bir şekilde hesaplanan sonucu bir sonraki nörona aktarır. Ağda bulunan her bir nöron, ağın başında verilen ham görüntü piksellerini işleyerek ağın sonunda bir sınıfa ait skor değerini hesaplamaktadır. Hesaplanan değer en son tamamen bağlı (fully connected) katmanda işlenerek görüntünün hangi sınıfa ait olduğu bilgisini veren tahmin değeri üretilir. 1998 yılında Yann LeCun ve arkadaşları [21] tarafından yayınlanmış ve ilk başarılı sonucu veren çok katmanlı yapay sinir ağı modelini Şekil 1.4'te görmek mümkündür. Posta numaraları ve banka çekleri üzerindeki sayıların sınıflandırılması ve okunması için geliştirilmiştir.



Şekil 1.4. LeNET-5 Evrişimsel Sinir Ağı Mimarisi.[21]

Şekil 1.4'de görüldüğü gibi LeNET-5, her biri eğitilebilir ağırlık parametrelerine sahip olan toplam 7 katmandan oluşmaktadır. Verilen giriş görüntüsünün piksel değerleri normalize edilerek görüntünün arka planı beyaz, ön planı ise siyah hale getirilerek sisteme verilen giriş görüntülerinin piksel değerleri 0 ve 1 düzeyine indirgenmiştir. Bu sayede

giriş değeri yaklaşık 0 ve varyans değeri yaklaşık 1 yapılarak öğrenme süresi hızlandırılmıştır.

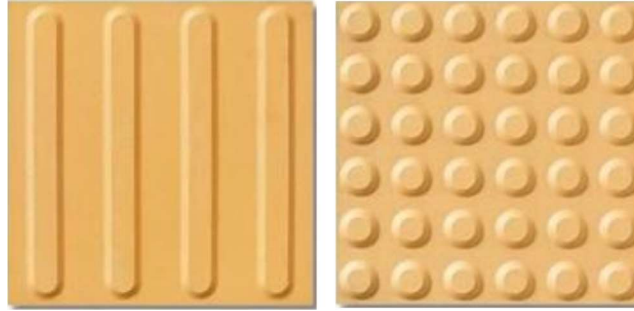
1.1.3. You Only Looked Once (YOLO)

YOLO-V3 [22]ağı, YOLO[23] ve YOLO-V2 [24]ağları kullanılarak geliştirilmiştir. Çalışma prensibi olarak YOLO ağı, tespit etme problemini regresyon problemine dönüştürmektedir. YOLO geleneksel yöntemlerden olan Hızlı R-CNN[25] in aksine nesne tespit etmede nesnenin bulunabileceği muhtemel alanlara ihtiyaç duymamaktadır. Regresyon sayesinde görüntüde bulunan her sınıf için sınırlayıcı kutu(Bounding Box) koordinatlarını ve olasılıklarını aynı anda oluşturabilmektedir. Her bir sınıf için görüntüde ayrı ayrı işlem yapmak yerine görüntüye yalnızca bir kere bakılır ve tüm sınıflara ait sınırlayıcı kutu koordinatları ve olasılıkları oluşturulur. Bu sayede görüntü üzerinde hangi nesnelerin olduğunu ve nesnelerin görüntünün tam olarak neresinde olduğunu öğrenmek oldukça hızlı bir şekilde gerçekleştirilmektedir. Bu durum geleneksel nesne tespiti yapan sistemlerle karşılaştırıldığında tespit süresini oldukça kısaltmaktadır. YOLO'nun çalışma prensibi ve konfigürasyonu Bölüm 2'de detaylı bir şekilde açıklanmıştır.

1.2. Dokunsal Parke Yüzeyleri

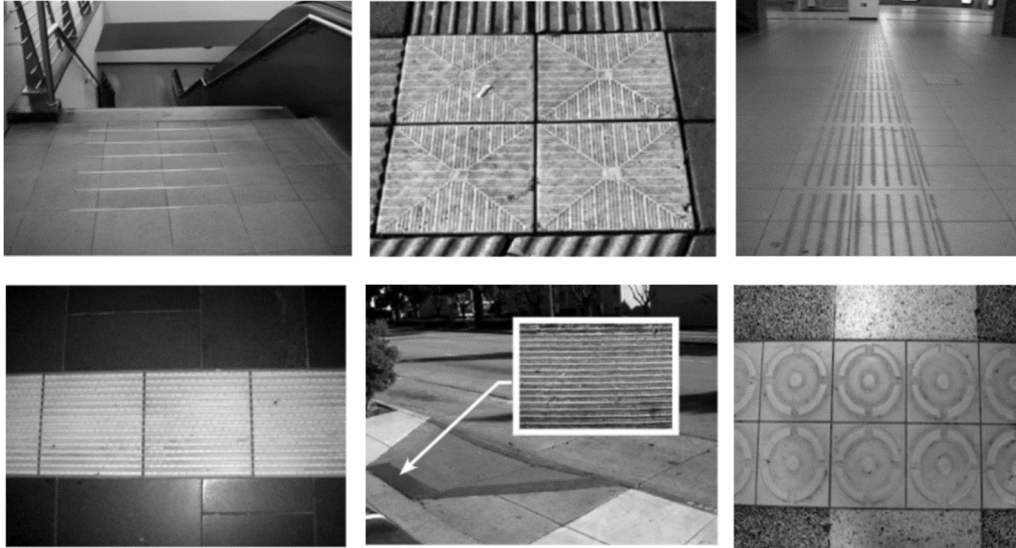
Dokunsal parke yüzeyleri, gerek iç mekânlarda (metro, alışveriş merkezi, havaalanı, kütüphane) gerekse dış mekânlarda (yaya kaldırımı, park ve bahçeler) görme engelli bireylerin bir yerden başka bir yere giderken daha rahat hareket edebilmeleri amacıyla zemine döşenmiş kabartmalı yüzeylerdir. İlk olarak 1965 yılında Japonya'da icat edilen dokunsal parke yüzeyleri günümüzde tüm Dünyada yaygın bir şekilde kullanılmaktadır [26]. Şekil 1.1'de dokunsal parke yüzeylerini görmek mümkündür.

Farklı ülkelerde, dokunsal parke, kesilmiş kubbe, tespit edilebilir uyarı, dokunsal zemin yüzeyi göstergesi ve dokunsal kılavuz yolu gibi farklı isimlere sahiptir [2].



Şekil 1.5. Dokunsal Parke Yüzeyleri.[26]

Şekil 1.6’da farklı ülkelerde kullanılan dokunsal parke yüzeyleri görülmektedir [26]. Dokunsal parkeleri, kısmen veya tamamen görme yetisini kaybetmiş kişiler kullanmaktadır. Hem iç mekânlarda hem de dış mekânlarda bulunan yürüyüş yollarında ekstra herhangi bir değişikliğe ihtiyaç duyulmadan düşük bir maliyetle kurulabilmesi sayesinde, dokunsal parke yüzeyleri görme engelli bireyleri yönlendirmede en etkili sistem olarak kabul edilmektedir [26]. Şekil 1.5’te görüldüğü gibi: tehlike alanlarını veya varış yerlerinin yerini belirten uyarı blokları ve hareket yönünü gösteren yön blokları olarak iki tip dokunsal parke yüzeyi bulunmaktadır. Dokunsal parke yüzeyleri, görme



Şekil 1.6. Farklı Ülkelere Göre Dokunsal Parke Yüzeyleri.[26]

engelli kişilerin kendi konumlarını, tehlike konumlarını ve yürüme yönlerini anlayabilmelerini sağlamak için tasarlanmıştır. Bu nedenle, dokunsal parke yüzeylerinin doğru ve güvenli bir şekilde kurulması hayati önem arz etmektedir. Buna rağmen, Türkiye’de yapılan yanlış ve tehlikeli dokunsal parke yüzeyi kurulumları görme engelli bireylere büyük engel ve tehlike oluşturmaktadır. Bu gibi durumları yurtdışında da görmek mümkündür [2]. Şekil 1.7’de yanlış DPY kurulumlarına ait görseller görülmektedir. Bu tip yapılan yanlış kurulumlar nedeniyle görme engelli bireylerin hareket kabiliyeti kısıtlanmakta ve yanlış bilgilendirme yapılabilmektedir. Bu tip hataların daha kontrollü bir şekilde algılanıp ilgili birimlere haber verilmesi, hataların düzeltilmesi açısından oldukça önemlidir. Görme engelli bireyler, bu tip hataların konumlarını doğru bir şekilde yetkililere bildirmesi oldukça zor bir durumdur. Bu tip hataların otonom bir şekilde tespit edilip harita konum bilgisine göre yetkili kişilere bildirilmesi, olası kazaların engellenmesi ve görme engelli bireylerin doğru ve güvenli



Şekil.1.7. Yanlış Kurulum Yapılan Dokunsal Parke Yüzeyleri.[26]

bir şekilde istedikleri yere seyahat etmeleri açısından oldukça önemlidir. Hem bu amaç doğrultusunda hem de görme engelli bireylerin otonom bir şekilde seyahat edebilmeleri amacıyla yapılması planlanan çalışma için gerekli olan sistemin temelleri bu tez çalışmasında atılmıştır.

1.3. Amaç ve Önem

Bu çalışmada hem görme bozukluğuna sahip bireylere yardımcı teknolojilerde kullanılabilecek hem de otonom bir şekilde hareket edebilen araçlarda kullanılabilecek görüntülerde, videolarda ve gerçek zamanlı çalışan sistemlerde dokunsal parke yüzeylerini tespit edebilen bir model gerçekleştirilmesi amaçlanmıştır. Derin öğrenme yöntemlerinin görüntü işleme alanındaki uygulanabilirliği ve uygunluğu göz önüne alınarak çalışmada temel olarak derin öğrenme yöntemleri ve görüntü işleme algoritmaları kullanılmıştır. Derin öğrenmeye dayalı nesne tespiti söz konusu olduğunda karşılaşılan üç temel nesne tanıma metodu vardır. Bunlar; Region-Convolutional Neural Network (R-CNN), Single Shot MultiBox Detector (SSD) ve You Only Look Once(YOLO) dur[23,25,27]. Yapılan çalışmada YOLO metodu ve evrimsel sinir ağıları [28] kullanılmıştır. Veri seti olarak bu çalışma kapsamında araştırmacılar tarafından oluşturulan ve başka araştırmacıların erişimine sunulan 4580 etiketli görüntüden oluşan MDPY(Marmara Dokunsal Parke Yüzeyi) veri seti kullanılmıştır. Veri seti üzerinde yapılan deneylerde %89 F1-Skor, %92 ortalama hassasiyet ve %81 IoU değeri ile dokunsal parke yüzeyleri tespiti gerçekleştirilmiştir.

Yapılan Katkılar:

- Gerçekleştirilen çalışmada; bu alanda yeni ve büyük veri seti olan Marmara Dokunsal Parke Yüzeyleri (MDPY) oluşturulmuş ve bu veri seti üzerinde YOLO yöntemi kullanılarak, belirli bir kamera açısına bağlı kalmadan dokunsal parke yüzeyleri tespiti gerçekleştirilmiştir.
- Yüksek performans değerlerine sahip yeni bir model geliştirilmiştir.

- Derin öğrenme yöntemlerinin dokunsal parke yüzeyi tespit etme alanında kullanılabilirliği kanıtlanmıştır.
- Dokunsal parke yüzeyi tespit etme alanında elde edilen yüksek hız ve yüksek doğruluk değerleri, gerçek zamanlı çalışan sistemlerde önerilen sistemi kullanım imkânı sağlamıştır.

1.4. Literatür Araştırması

Görme engelli bireylerin seyahat ederken engelleri algılaması ve korunması zordur. Teknolojinin gelişmesiyle birlikte engelleri önceden algılayabilen ve engelli bireyleri yönlendiren sistemler tasarlanmıştır. Bu başlık altında derin öğrenme yöntemleri kullanılarak yapılan görüntü işleme çalışmaları ve engel algılama ve görme engelli bireyler için tasarlanmış yardımcı sistemler ile ilgili literatürdeki önemli çalışmalara değinilmektedir.

1.4.1. Görüntü İşleme İle İlgili Çalışmalar

Derin öğrenme günümüzde oldukça popüler bir çalışma alanıdır. Doğal dil işleme, görüntü işleme, ses işleme, nesne tanıma ve yorumlama, ilaç üretimi, astronomi gibi alanlarda yaygın bir şekilde kullanılmaktadır.

Jun Xiao ve arkadaşları [29] elektromanyetik tomografideki görüntüleme problemleri üzerine yaptıkları çalışmada, iki adet farklı derin öğrenme tabanlı algoritma geliştirmişlerdir. Bunlardan birincisi seyrek oto kodlayıcı(SAE) ve radyal temel fonksiyonlu (RBF) ağırları birleştirerek oluşturulan algoritmadır. İkincisi ise optimize edilmiş, yığın normalizasyonu(Batch Normalisation) ve aktivasyon fonksiyonu kullanan tam bağlı (Fully Connected) ağ yapısı temelli algoritmadır. Geliştirilen bu iki algoritma ve geleneksel görüntüleme algoritmaları örnekler üzerinde uygulanmış ve sonuçlar karşılaştırılmıştır. Bulunan sonuçlara göre geliştirilen algoritmalar GPU üzerinde çalıştıklarından geleneksel yöntemlere göre hem çok daha hızlı çalışmakta hem de çok daha iyi bir görüntü sağlamaktadırlar.

Kuasarlar, konakçı gökadalara dinamik olan merkezlerindeki süper kütleli kara delikler üzerinde, geniş bir frekans aralığında yüksek parlaklıkların oluşmasıdır[30]. Pasquet-Itam, J ve Pasquet, J [30] yaptıkları çalışmada kuasarların tespit edilmesi için evrişimsel sinir ağlarını kullanmışlardır. New Mexico da bulunan Apache Point gözlem evindeki 2.5-metre teleskop ile elde edilen çoklu filtreli görüntüleme ve spektroskopik kırmızıya kayma(redshift) araştırmasında elde edilen verileri kullanmışlardır. Bu veriler üzerinde evrişimsel sinir ağı(Convolutional Neural Network) kullanılarak, kuasarların fotometrik ışımlarını tahmin etme, teşhis etme ve sınıflandırma işlemleri yapılmıştır. Elde edilen sonuçlar diğer sınıflandırma ve kümeleme yöntemleri olan rastgele orman sınıflayıcı(random forest classifier), k en yakın komşu (K-nearest-neighbours), destek vektör makinesi(support vector machine) ve gauss işlem sınıflandırıcı (Gaussian process classifier) ile karşılaştırılmıştır. Evrişimsel sinir ağları diğer sınıflandırma yöntemlerine göre çok daha yüksek doğrulukta sonuçlar vermiş ve 175 yeni kuasar adayı tespit etmiştir.

Katsumi Hagita ve arkadaşları[31] derin öğrenme tabanlı olarak çalışabilen Çekişmeli Üretici Ağlar ile Süper Çözünürlük(Super Resolution with Generative Adversarial Networks SRGAN) modelini kullanarak görüntünün daha yüksek ve eşit çözünürlükte görüntülenmesini sağlamışlardır. Verileri elde etmek için odaklanmış iyon demeti ile donatılmış taramalı elektron mikroskopu (Scanning electron microscopy equipped with a focused ion beam FIB-SEM) kullanılmıştır. FIB-SEM'in çalışma mantığında, numune yüzeyi bir iyon ışınıyla soyulur ve FIB'ye ortogonal olarak yerleştirilmiş bir SEM ile görüntülenir[31]. Yanal çözünürlük SEM tarafından yönetilirken, derinlik çözünürlüğü, yani FIB frezeleme yönü, soyulmuş ince tabakanın kalınlığı ile belirlenir[31]. Genellikle yanal çözünürlük derinlik çözünürlüğünden daha iyi olmaktadır. Bu durum 3D görüntüleme yapıldığı zaman asimetrik bir çözünürlüğün ortaya çıkmasına mahal vermektedir[31]. Bu problemi düzeltmek için Katsumi Hagita ve arkadaşları[31] Çekişmeli Üretici Ağlar ile Süper Çözünürlük(Super Resolution with Generative Adversarial Networks SRGAN) modelini kullanarak görüntünün daha yüksek ve eşit çözünürlükte görüntülenmesini sağlamışlardır. Yapılan çalışmanın sonucunda SRGAN metodu, geleneksel yöntemlerden olan iki çizgili(bilinear) ve bikübik(bicubic) metotları

ile karşılaştırılmıştır. 8 bitlik görüntüler üzerinde yapılan çalışmada SRGAN metodunun daha iyi olduğu ve bu metodu sırasıyla bikübik ve iki çizgili metotların takip ettiği gözlemlenmiştir. 2 bitlik görüntüler üzerinde çalışma yapıldığında ise geleneksel yöntemlerin daha iyi olduğu ve sırayı tersine çevirdiği gözlemlenmiştir.

En değerli görsel kaynaklardan ve referanslardan biri olduğu için biyomedikal görüntüler, biyomedikal analiz ve hastalıkların tanı ve tedavisi alanlarında hekimler ve biyomedikal araştırmacılar için önemli bir rol oynamaktadır[32,33]. Aynı zamanda, biyomedikal görüntü teknolojisinin geniş uygulama alanlarından dolayı [34], X ışını, ultrason, manyetik rezonans görüntüleme ve bilgisayarlı tomografi gibi çok sayıda biyomedikal görüntüleme cihazı tarafından çok çeşitli görüntüler üretilmekte ve farklı formatlarda saklanmaktadır. Tüm bu farklı formatta olan verileri depolamak ve istenildiğinde tekrardan geriye almak oldukça zor bir iştir. Biyomedikal olmayan görüntüler için tasarlanan geleneksel biyomedikal görüntü alma yöntemleri ve içerik tabanlı görüntü alma yöntemleri (content-based image retrieval CBIR), yalnızca bir görüntüyü tanımlamak için piksel ve düşük düzey özellikleri(renk, şekil, vb.) kullanmakta veya görüntüleri tanımlamak için derin özellikleri kullanmamaktadır[35]. Fakat biyomedikal görüntüleri bu şekilde depolamak ve geri almaktan çok daha verimli yöntemleri geliştirmek mümkündür. Shuchao Pang ve arkadaşları [35] geleneksel olan bu yaklaşımlardan farklı olarak derin tercih öğrenme metotları ile biyomedikal görüntüleri indeksleme ve geri alma sistemi tasarlamışlardır. Tasarlanan bu sistemde biyomedikal görüntülerin ayırt edici özelliklerini göstermek için yığılmış özerk kodlayıcılar (stacked denoising autoencoders SDAE) ve evrişimli sinir ağları(convolutional neural networks CNN) kullanılmaktadır. Aynı zamanda biyomedikal görüntü veri tabanındaki görüntülerin benzerlik katsayılarını çıkarabilmek için eğitme ve öğretme modelli olarak çalışan tercihli öğrenme teknolojileri Shuchao Pang ve arkadaşları [35] tarafından kullanılmıştır. Yaptıkları çalışma ile ilk kez biyomedikal görüntü alma için tercihli öğrenme teknolojisi kullanılmıştır[35]. Oluşturulan iki güçlü algoritma, diğer popüler biyomedikal görüntü indeksleme yaklaşımları ve var olan görüntü alma metotları ile detaylı bir şekilde herkesin erişimine açık olan çeşitli görüntü veri tabanları üzerinde

karşılaştırılmıştır. Karşılaştırmanın sonucunda görülmektedir ki Shuchao Pang ve arkadaşları [35] tarafından yapılan algoritmalar, en modern biyomedikal görüntü indeksleme tekniklerini bile geride bırakmıştır.

Videolardaki insan davranışlarının sınıflandırılması bilgisayarlı görü alanında oldukça popüler bir konudur.[36]. Özellikle son yıllarda, otomatik video gözetiminde anormal insan davranışlarının tespit edilmesi konusu üzerine birçok çalışma yapılmıştır[37,38]. Manassés Ribeiro ve arkadaşları [39] bu alanda çalışma yaparak, videolardaki anormal insan davranışlarını tespit etmeyi amaçlamışlardır. Son zamanlarda evrimsel sinir ağları nesne tanıma alanında son teknoloji olmayı başarmıştır[40,41]. İnsan yardımı ile görüntü ayırıcılar ile karşılaştırıldığında evrimsel sinir ağları, hem görüntü özelliklerini otomatik olarak öğrenebildiği için hem de görüntü gösteriminde üstün bir şekilde ayırt edici güç olduğu için tercih edilmekte ve son teknoloji olma unvanını korumaktadır[42,43]. Fakat evrimsel sinir ağları denetimli bir şekilde eğitilmektedir. Yalnızca normal davranışların olduğu bir veri seti ile eğitildikleri takdirde anormal davranışların tespitini yapmaları güçleşmektedir. Bu sorunla başa çıkabilmek için oto-kodlayıcılar devreye girmektedir. Oto-kodlayıcılar ilk başlarda görüntü alma konusu üzerine çalışmasına rağmen son zamanlarda video anormallik tespiti alanında da çalışmaktadırlar[38,43]. Oto-kodlayıcılar sadece bir tek sınıfı sınıflandırma problemleri ile ilgilidirler. Yalnızca bir sınıfla eğitilebilmektedirler. Fakat oto-kodlayıcılar giriş verisi olarak bir boyutlu vektör aldıklarından görüntü ve video dizilerindeki iki boyutlu dizileri yakalayamamaktadırlar. Bu sorunu gidermek için Manassés Ribeiro ve arkadaşları [39] evrimsel oto-kodlayıcıları kullanmışlardır. UCSD yaya veri seti [44] ve Avenue veri seti [45] kullanılmıştır. UCSD veri seti içerisinde 2 alt veri içermektedir. Bunlardan birinci alt veri seti içerisinde 5500 adet normal ve 3400 adet anormal olan video çerçevesi bulunmaktadır. İkinci alt veri seti içerisinde yaklaşık 346 adet normal ve 1652 adet anormal olan video çerçevesi bulunmaktadır. Orjinal video çerçeveleri ile görünüm ve hareket özelliklerinin kombinasyonları evrimli oto-kodlayıcılara giriş verisi olarak verilmiştir. Rekonstrüksiyon hatalarının düzenli hale getirilmesi(Regularization of reconstruction errors, RRE.), çerçevelerin anormallik seviyesini ölçmek için

kullanılmıştır. Sınıflandırma performansı, farklı eşik değerlerinde ROC eğrisinin altındaki alan (area under the ROC curve, AUC) kullanılarak değerlendirilmiştir. Değerlendirme sonucunda UCSD-2 yaya veri setinde elde edilen en yüksek AUC değeri 0.847 ve Avenue yaya veri setinde elde edilen en yüksek AUC değeri 0.772 olarak elde edilmiştir. Kolmogorov [46] karmaşıklığı, video mekânsal karmaşıklığının bir ölçütü olarak kullanılmıştır. Yöntemleri doğrulamak için deneyler, bahsedilen açık kaynaklı veri setleri üzerinde kullanılmıştır. Çalışma sonucunda Manassés Ribeiro ve arkadaşları [39] kendi algoritmalarından daha genel ve biçimsel bir şekilde normallik/anormallik ayrımını tanımlayabilen yöntemin olmadığını ileri sürmektedirler.

Jiaquan Shen ve arkadaşları [47] yaptıkları çalışmada derin evrişimli ağlardaki hiper özellik haritasını kullanarak havadan çekilmiş görüntülerdeki araçları tespit etmeyi amaçlamışlardır. Yapılan algoritmanın temelini Faster R-CNN[48] algoritması oluşturmaktadır. Bu algoritma temel olarak doğal sahnelerdeki nesneleri tespit etmektedir. Havadan çekilmiş görüntülerdeki küçük nesneleri tespit etmek için uygun değildir. Jiaquan Shen ve arkadaşları, havadan çekilmiş görüntülerdeki küçük araçları doğru ve etkili bir şekilde tespit etmek için Elwise modeli ve Concat modelini hiperaktif özellik harita ağı ile birleştirerek yeni bir yaklaşım sunmaktadır. Tasarlanan algoritmanın tespit etme performansını Munich veri seti ve kendi oluşturdukları veri seti üzerinde değerlendirmişlerdir. Yapılan sistem 82.2% duyarlılık (Recall) ve 90.2% doğruluk oranına sahiptir.

1.4.2. Dokunsal Parke Yüzeyi Tespiti İle İlgili Çalışmalar

Marcelo C. Ghilardi ve arkadaşları [4] görme engelli bireylere yardımcı olmak amacıyla gerçek zamanlı olarak dokunsal parke yüzeylerinin tespitini gerçekleştirmişlerdir. Yaptıkları çalışmada Canny, Laplace ve Sobel gibi kenar belirleme algoritmalarını ve karar ağaçlarını birleştirerek yeni bir yaklaşım elde etmişlerdir. Araştırmacılar tarafından oluşturulan veri setinde 521 adet görüntü bulunmaktadır. Görüntüler, bir metre yükseklikte ve yer ile 45 derece açı yapacak şekilde kişinin bel bölgesine bağlanmış olan akıllı telefon ile elde edilmiştir. Bu görüntülerin 320 âdetinde

dokunsal parke yüzeyi mevcutken 201 âdetinde dokunsal parke yüzeyi bulunmamaktadır. Yapılan çalışmada sistem yaklaşık olarak 16.27 fps. ile çalışmaktadır. Yapılan sistemin doğruluk oranı yaklaşık olarak %88.48 oranına sahiptir.

Takuya Asami ve Kengo Ohnishi [3] benzer bir çalışmada, görme engelli bireylerin yaya geçidini bulmasına yardımcı olan sistemi gerçekleştirmişlerdir. Sistem gözlüğe entegre edilmiş bir usb kamera, bileğe takılan bir adet titreşimli bileklik, görme engelli bastonuna entegre edilmiş görme engelli alfabesinde uyarı veren bir bildirim sistemi ve usb kameradan gelen verileri işlemek için kullanılan bir adet bilgisayar olmak üzere dört parçadan oluşmaktadır. Bu çalışmada, usb kameradan alınan görüntülerde normalize edilmiş çapraz korelasyon yöntemi ile yaya geçidi algılanarak, kullanıcıya yönlendirme yapan bir sistem geliştirilmiştir. Yaya geçidi algılandığında kullanıcının bilekliği titreterek kullanıcının yaya geçidinin önünde durması sağlanmıştır. Geliştirilen sistem %86.7 doğrulukla yaya geçitlerini tespit etmektedir.

Daniel Centeno Einloft ve arkadaşları [49] DPY tespiti için yapılan çalışmaya [4] benzer olarak alışveriş merkezî, metro ve otobüs terminalleri gibi iç mekânlarda kullanılabilen DPY tespiti yapan sistemi yapmayı amaçlamışlardır. Çalışmalarında görüntü sınıflandırmada kullanılan gri seviyeli eş-oluşum matrisi metodunu ve denetimli öğrenme yöntemlerinden olan destek vektör makinelerini kullanmışlardır. İlk sonuçlar umut vaad etmesine rağmen çalışma doğruluğu ile ilgili herhangi bir sayısal değer verilmemiştir.

Xu Jie ve arkadaşları[50] yaptıkları çalışmada görüntü işleme algoritmalarını kullanarak dokunsal parke yüzeylerini tespit etmeyi amaçlamışlardır. Yapılan sistemde çok renkli görüntü segmentasyonu ve kirsch kenar belirleme algoritması kullanılmıştır. Hough dönüşüm oryantasyonu kullanarak dokunsal parke yüzeyini şerit olarak elde etmişlerdir. 2 fps çalışma hızına sahip olan ve portatif olarak kullanılabilen sistemin, çalışma doğruluğu ile ilgili herhangi bir sayısal değere değinilmemiştir.

A.M. Kassim ve arkadaşları [51] görme engelli bireylerin dokunsal parke yüzeylerini tespit edebilmeleri için yeni bir baston tasarlamışlardır. Tasarlanan bastona

eklenen web kamera ile görüntü alınmakta ve arduino Xbee verici ile bilgisayara gönderilmektedir. Bilgisayar ortamında Matlab programı ile görüntü farklı filtrelerden geçirilerek işlenmektedir. İşlenen görüntülerden elde edilen “A” alan değeri $m = 4\pi A/r^2$ formülünde kullanılarak m metrik değeri hesaplanmaktadır. Hesaplanan m değerine göre alınan görüntünün dairesel veya doğrusal olduğuna karar verilmektedir. $0.15 < m < 0.30$ aralığında alınan görüntünün doğrusal bir görüntü olduğu yani kullanıcının ilerleyebileceği, $0.85 < m < 1.0$ aralığında alınan görüntünün dairesel yani uyarı parke yüzeyi olduğu kullanıcının durması gerektiği anlaşılmaktadır. Hesaplanan m metrik değerine göre alınan sonuçlar arduino Xbee verici ile tekrar bastona gönderilmektedir. Gelen verilere göre sisteme entegre olan kulaklık ile kullanıcı bilgilendirilmektedir. Gerçekleştirilen sistem belirli bir donanım ekipmanına sahip olmayı gerektirdiği için pratik bir kullanıma sahip değildir.

Literatür taraması sonucunda daha önce yapılan sistemlerin günlük hayatta kullanılabilirliğinin oldukça zor olduğu anlaşılmaktadır. Bu zorluklar; düşük fps değerine sahip sistemlerin[3,4,50] gerçek zamanlı olarak kullanılamaması, hem iç mekânlarda hem de dış mekânlarda yalnızca bir sistem [4,49] ile dokunsal parke yüzeylerinin tespit edilememesi, alışveriş merkezi, metro, vb. gibi iç mekânlar için ayrı, yürüyüş yolu, cadde gibi dış mekânlar için ayrı bir sistemin oluşturulması, [3,51] kullanıcının sistemi kullanmak için sırt çantasında bilgisayar, baston, bileklik gibi ekstra donanımlar kullanmak zorunda olması, [49] veri azlığı nedeni ile çalışmaların tamamlanamaması gibi sorunlardan oluşmaktadır. Literatürde bulunan çalışmaların aksine bu çalışmada yapılan sistem ile saniyede 60 kare ile işlem yaparak gerçek zamanlı olarak çalışılmış ve baston, bileklik, bilgisayar gibi ekstra donanımlara ihtiyaç duymadan hem iç mekânlarda hem de dış mekânlarda dokunsal parke yüzeylerinin tespiti yapılmıştır. Araştırmacılar tarafından oluşturulan ve üzerinde çalışılan veri seti sayesinde veri azlığı nedeniyle oluşan problemlerin önüne geçilmiştir.

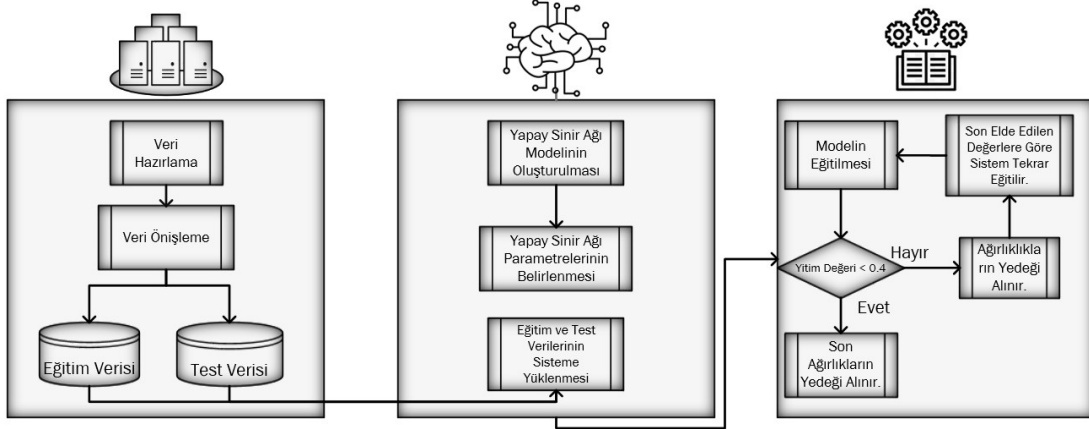
Tablo 1.1’de dokunsal parke yüzeyi tespiti ile ilgili yapılan başlıca çalışmaların kullandığı yöntemler, doğruluk değerleri, avantajları ve dezavantajlarını görmek mümkündür.

Tablo 1.1. Dokunsal Parke Yüzeyi Tespiti İle İlgili Literatürde Bulunan Çalışmalar

Çalışma	Kullanılan Yöntem	Doğruluk	fps	Avantajlar	Dezavantajlar
[4]	Canny, Laplace, Sobel Kenar Belirleme Algoritmaları ve Karar Ağaçları	%88.48	16.27	Geleneksel görüntü işleme algoritmalarına ait örnekler mevcuttur.	Düşük fps değeri gerçek zamanlı sistemlerde kullanıma uygun değildir.
[49]	Gri seviyeli eş-oluşum matrisleri ve Destek Vektör Makineleri	-	-	Danışmanlı öğrenme yöntemlerine ait örnekler mevcuttur.	Sistemin çıktıları ile ilgili herhangi bir bilgi verilmemiştir.
[50]	Çok Renkli Görüntü Segmentasyonu ve Kirsch Kenar Belirleme Algoritması.	-	2	Geleneksel görüntü işleme algoritmaları ve donanımsal parçaların bir arada kullanılmasına ait örnekler mevcuttur.	Düşük fps değerine sahiptir. Günlük hayatta kullanımı oldukça masraflı ve zor bir sistemdir.

2. MATERYAL VE YÖNTEM

Gerçekleştirilen tez çalışması Şekil.2.1’de görüldüğü gibi temel olarak 3 adımdan oluşmaktadır. Birinci adımda veri setinin hazırlanması ile ilgili alt işlemler, ikinci adımda derin öğrenme için kullanılacak evrimsel sinir ağının oluşturulması ve konfigürasyonu, üçüncü adımda sistemin eğitilmesi ve diğer sistemlerde de kullanılmak üzere evrimsel sinir ağı ağırlıklarının elde edilmesi gerçekleştirilmiştir.



Şekil 2.1 Sistem Mimarisi.

2.1. Kullanılan Veri Seti

Yapılan çalışmada araştırmacılar tarafından oluşturulan veri seti kullanılmıştır. Veri hazırlama ve önleme evresinde, iç ve dış ortamlarda çekilen görüntü ve video gibi görsel kaynaklar belirli işlemlerden geçirilerek etiketli bir veri seti haline getirilmiştir.



Şekil 2.2.İç Mekânlara Ait Veri Setinde Bulunan Örnek Görüntüler.



Şekil 2.3. Dış Mekânlara Ait Veri Setinde Bulunan Örnek Görüntüler.

Görüntüler İstanbul'un çeşitli semt ve bölgelerinde bulunan kütüphane, metro, parklar, yürüyüş yolları ve caddeler gibi gündelik hayatta sıkça kullanılan yerlerden temin edilmiştir. Görüntüler temin edilirken GoPro Hero 7 aksiyon kamerası ve Sony HDR-X3000R-4K aksiyon kamerası kullanılmıştır. Görüntü temini sırasında herhangi bir kişinin özel hayatı ihlal edilmemiştir. Elde edilen iç mekân görüntülerine Şekil.1.9'da, dış mekân görüntülerini Şekil.1.10'da görmek mümkündür.

2.1.1. Veri Çoğullama

Elde edilen görüntüleri çok daha hızlı ve sisteme uygun bir şekilde kullanmak için yeniden boyutlandırma işlemi yapılmış ve 1024x768 boyutuna getirilmiştir. Evrişimsel sinir ağları ile danışmanlı öğrenmede, toplanması ve hazırlanması oldukça maliyetli olan çok sayıda etiketli veriye ihtiyaç duyulmaktadır[52].İhtiyaç duyulan veriyi elde etmek amacıyla python programlama dili ve OpenCV kütüphanesi kullanılarak veri artırma işlemi gerçekleştirilmiştir. Başlangıçta 1200 adet görüntü bulunan veri setine veri artırma işlemleri uygulanarak görüntü sayısı 4580'e çıkarılmıştır. Veri artırma yöntemlerinden aynalama, beyazlatma, döndürme, vb. yöntemler kullanılmıştır[53]. Görüntülerin büyük

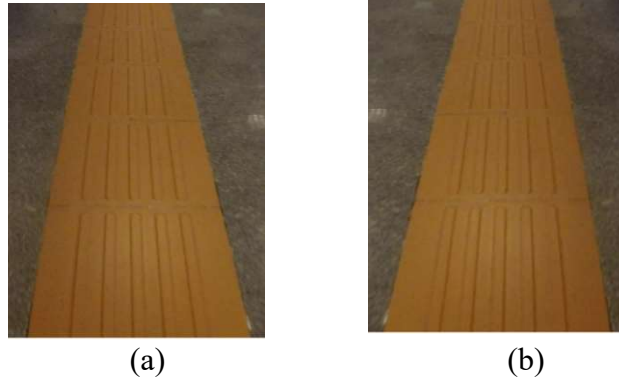


Şekil 2.4. Veri Çoğullama Örnek Görüntüler.

bir kısmında birden fazla DPY olması, etiketli veri sayısını daha da artırmaktadır. Şekil 2.4'te veri çoğaltma yapılan örnek görüntüleri görmek mümkündür.

2.1.2. Veri Temizleme

Veri çoğullama işleminden sonra elde edilen veriler kontrol edilerek kendini tekrar eden, günlük hayatta karşılaşamayacağımız veriler veri setine dâhil edilmemiştir. Kendini tekrar eden verilerin olması yapay sinir ağı modelinin eğitimi sırasında verilen veri setini ezberleyerek(overfitting) veri seti dışında herhangi bir veri verildiğinde tespit işleminin yapılamamasına neden olmaktadır. Bunun önüne geçmek için veri seti kontrol edilerek uygun olmayan görüntüler veri setine dâhil edilmemiştir.



Şekil 2.5. (a) Veri setinde bulunan örnek görüntü. (b) Çoğullama işleminden sonra elde edilen örnek görüntü.

Şekil 2.5 (a)'da veri setine dâhil edilmiş bir görüntü bulunmaktadır. (b)'de veri çoğullama işleminde aynalama tekniği uygulandıktan sonra elde edilen görüntü bulunmaktadır. Bu iki görüntü arasında çok büyük bir fark bulunmamaktadır. Veri setinin eğitimi sırasında tekrara düşmemesi için ilgili görseller tespit edilerek veri setine dâhil edilmemiştir.

2.1.3. Veri Etiketleme

Elde edilen verileri, danışmanlı öğrenme yöntemi ile yapay sinir ağlarında kullanmak için etiketlemek gerekmektedir. Veri setinde bulunan görsellerin hangi kısımlarında DPY bulunduğu etiketlenerek etiketli veri seti oluşturulmuştur. Etiketleme işlemini yapmak

için python programlama dili ve pillow kütüphanesi kullanılarak yapılan LabelTool programı kullanılmıştır[54].



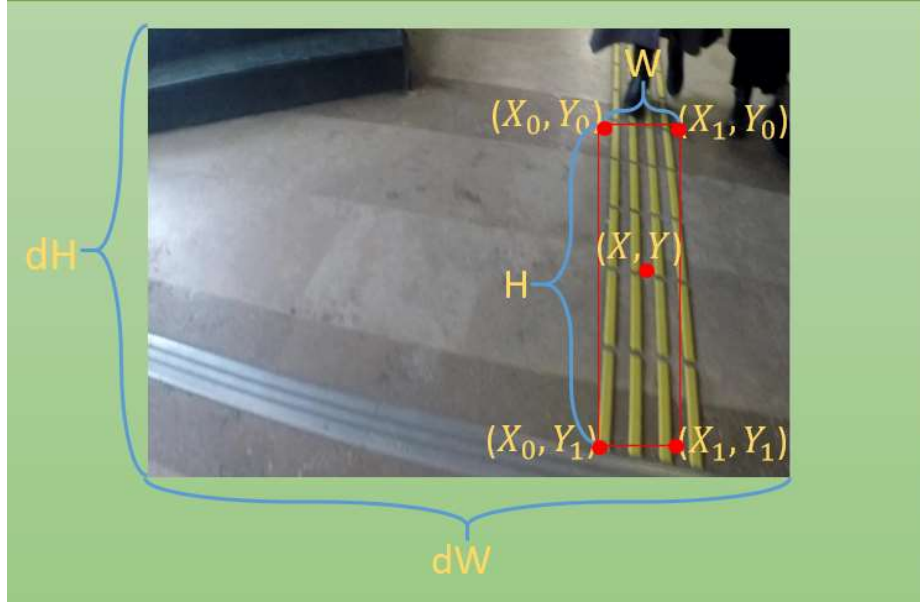
Şekil 2.6. LabelTool Veri Etiketleme Programı[54]

Şekil 2.6’da görüldüğü gibi görüntüler sınırlayıcı kutular ile etiketlenerek görüntü üzerinde nerede DPY olduğu görüntü adı ile aynı isimde bir text dosyası oluşturularak DPY’nin koordinatları kayıt edilmektedir. Bu işlem veri setinde bulunan 4580 verinin her biri için ayrı ayrı yapılmıştır.

2.1.4. Veri Setinin Normalize Edilmesi

Etiketlenen görüntülerin uzunluk(dH) ve genişliği(dW), DPY nesnesinin görüntünün tam olarak hangi konumunda olduğu bilgisini veren (X_0, Y_0, X_1, Y_1) noktaları ve görüntü üzerinde kaç adet DPY olduğu bilgisi text dosyası içerisine kayıt edilmektedir. Tüm görüntüler üzerinde bu işlem uygulanarak etiketli veri seti oluşturulmuştur. Oluşturulan veri setini, sistemde kullanılan YOLO [23] mimarisine uygun hale getirmek için normalizasyon işlemi yapılmaktadır. Normalizasyon işlemi ile (X_0, Y_0, X_1, Y_1) koordinatları 0 ve 1 aralığına indirgenmiştir. Bu işlem ile görüntü içerisindeki etiketli

DPY'nin, merkez noktası koordinatları(X,Y), yüksekliği(H) ve genişliği(W) bilgileri elde edilmektedir. Yapılan normalizasyon işleminin formülü Denklem 2.1'de verilmiştir.



Şekil 2.7. Etiketlenen Verinin Koordinatları.[23]

$$X = \frac{X_1 + X_0}{2} x \frac{1}{dW}$$

$$Y = \frac{Y_1 + Y_0}{2} x \frac{1}{dH}$$

$$W = (X_1 - X_0) x \frac{1}{dW}$$

$$H = (Y_1 - Y_0) x \frac{1}{dH} \quad (2.1)$$

X : Etiketli verinin orta noktasına ait X koordinatı.

Y : Etiketli verinin orta noktasına ait Y koordinatı.

W : Etiketli verinin genişliği

H : Etiketli verinin yüksekliği

2.1.5. Test ve Eğitim Sınıflarının Belirlenmesi

Denklem 2.1’de bulunan denklemlere göre hesaplanan yeni koordinat, yükseklik ve genişlik değerleri (X,Y,W,H) görüntünün adı ile aynı olan yeni bir text dosyasına kaydedilmektedir. Ön işlemler tamamlandıktan sonra veri seti, %65 eğitim ve %35 test verileri olmak üzere rastgele bir şekilde ikiye ayrılmıştır.

2.2. Kullanılan Yöntemler

Gerçekleştirilen sistem, YOLOV3-Dense, YOLO-V3 ve YOLO-V2 modelleri olmak üzere üç farklı model kullanılarak çalıştırılabilmektedir. YOLOV3-Dense modeli oluşturulurken YOLO-V3 modeli ve yoğun bağlantılı sinir ağı olan DenseNet ağı birleştirilerek yapılmıştır.

2.2.1. YOLO(You Only Look Once)

İnsanlar görüntülere bakarak görüntülerde hangi nesnelerin olduğunu, konumlarını birbiri ile olan etkileşimlerini anında anlayabilmektedir. İnsan görme sistemi oldukça hızlı ve doğru bir şekilde çalışmaktadır. Şekil 2.8’de görüldüğü gibi dikkatinin çok az bir kısmıyla bile araç kullanmak gibi karmaşık işleri yapabilmektedir. Nesne tespit etmede hızlı ve doğruluk oranı yüksek algoritmalar, bilgisayarların özel sensörler olmadan araba kullanmasına, yardımcı cihazların gerçek zamanlı sahne bilgilerini kullanıcılara iletimine

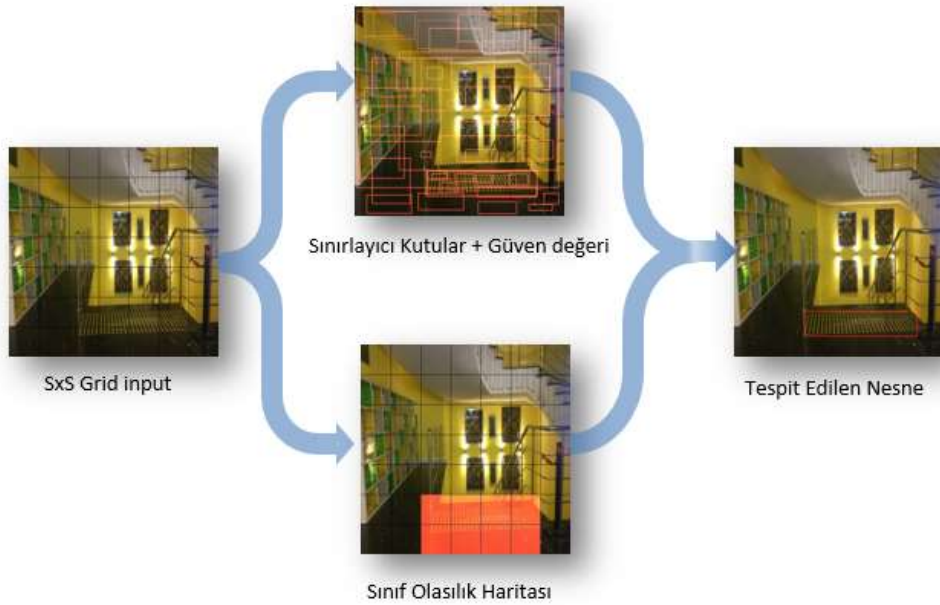


Şekil 2.8. Çoklu görev yapabilen bir insan.

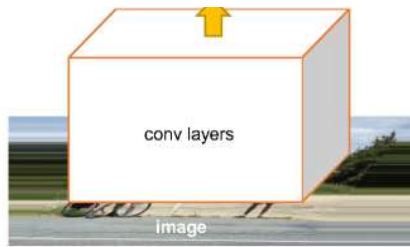
olanak sağlamıştır[23]. Geleneksel nesne tespit algoritmalarında tespit edilen nesneyi sınıflandırmak için ekstra bir sınıflandırma işlemi yapılmaktadır. Algoritmalar öncelikle bir görüntüde nesnelerin bulunması muhtemel olan alanları belirlemektedir. Belirlenen alanlarda sınıflandırıcı olarak tasarlanan evrimsel sinir ağları her bir bölge için ayrı ayrı çalıştırılarak nesne tespit edilmektedir. Yapılan sistemler iyi sonuçlar vermesine karşın görüntü iki ayrı işlemden geçirildiği için ortaya çıkan parametre sayıları ve ihtiyaç duyulan işlem gücü artmaktadır[55]. Bu nedenle yapılan sistemleri gerçek zamanlı çalışan sistemlerde kullanmak mümkün değildir. Bu şekilde tasarlanan en yeni yaklaşımlardan biri olan R-CNN[48] algoritmasında önce bir görüntüde bulunabilecek potansiyel sınırlayıcı kutular oluşturmakta ve ardından bu önerilen kutular üzerinde bir sınıflandırıcı çalıştırmak için bölge bazlı çalışan metotlar kullanılmaktadır. Sınıflandırmadan sonra, üretilen sınırlayıcı kutuların güven değerlerini iyileştirmek, aynı nesnenin birden fazla kez algılanmasını ortadan kaldırmak ve görüntüdeki diğer kutulara göre kutuları yeniden sıralama gibi işlemler yapılmaktadır[25]. Bu karmaşık işlemlerin her biri ayrı ayrı

eđitildiđinden sistem oldukça yavař ve optimize edilmesi zordur. řekil 2.9’da R-CNN mimarisini grmek mmkndr.

YOLO, geleneksel nesne tespit yntemlerinin aksine sınırlayıcı kutuların bulunması, sınıf olasılıklarının hesaplanması ve diđer tm iřlemleri tek bir regresyon problemi olarak ele almıř ve nesne tespit alanına yeni bir grř getirmiřtir. YOLO ile grntzerinde hangi nesnelerin nerede olduklarını tespit etmek iin grntye yalnızca bir kez bakılması yeterlidir(YOLO). Tek bir evriřimsel ađ ile aynı anda birden fazla sınırlayıcı kutu tahmin edilmekte ve her bir sınıf iin sınıf olasılıkları tahmin edilmektedir. Bu birleřik model, geleneksel nesne tespit etme yntemlerine gre eřitli faydalara sahiptir.



řekil 2.10. YOLO Tespit řeması.



řekil 2.9. Hızlı R-CNN Mimarisi.[48]

YOLO tespit modeli, eğitim setinde bulunan her bir görüntüyü Şekil.6’da gösterildiği gibi $S \times S$ ($S=7$) kare hücrelere(grid) bölmektedir. Tespit edilmesi istenen nesnelerden herhangi birinin merkez konumu bölünen hücrelerden herhangi birinin içerisinde ise, merkezin bulunduğu hücre o nesneyi tespit etmekten sorumludur. Her bir hücre, B sınırlayıcı kutularını üretmekte ve bu sınırlayıcı kutular için güven skorunu tahmin etmektedir. Güven skoru, modelin ürettiği sınırlayıcı kutunun bir nesne içerdiğinden ne kadar emin olduğunu ve üretilen sınırlayıcı kutunun üretilme olasılığının ne kadar doğru olduğunu yansıtmaktadır. Güven skoru = $\Pr(\text{Object}) * IOU_{pred}^{truth}$, $\Pr(\text{Object}) \in (0,1)$ olarak hesaplanmaktadır. Bölünen hücreler içerisinde herhangi bir nesne yok ise o hücrelerin güven skoru 0, var ise 1 olmaktadır. Her bir hücre aynı zamanda denklem 2.2’de görüldüğü gibi C sınıf koşullu olasılıklarını(conditional class probabilities) $\Pr(\text{Class}_i | \text{Object})$, tahmin etmektedir. Bu olasılıklar, nesne içeren hücreler ile bağlantılıdır. Üretilen sınırlayıcı kutu sayısına B bakılmaksızın her bir hücre başına sınıf olasılıklarının yalnızca biri tahmin edilmektedir. Test sırasında koşullu sınıf olasılıkları ve bireysel güven skoru tahminleri çarpılarak her bir kutu için sınıf bazında güven skorları hesaplanmaktadır.[23].

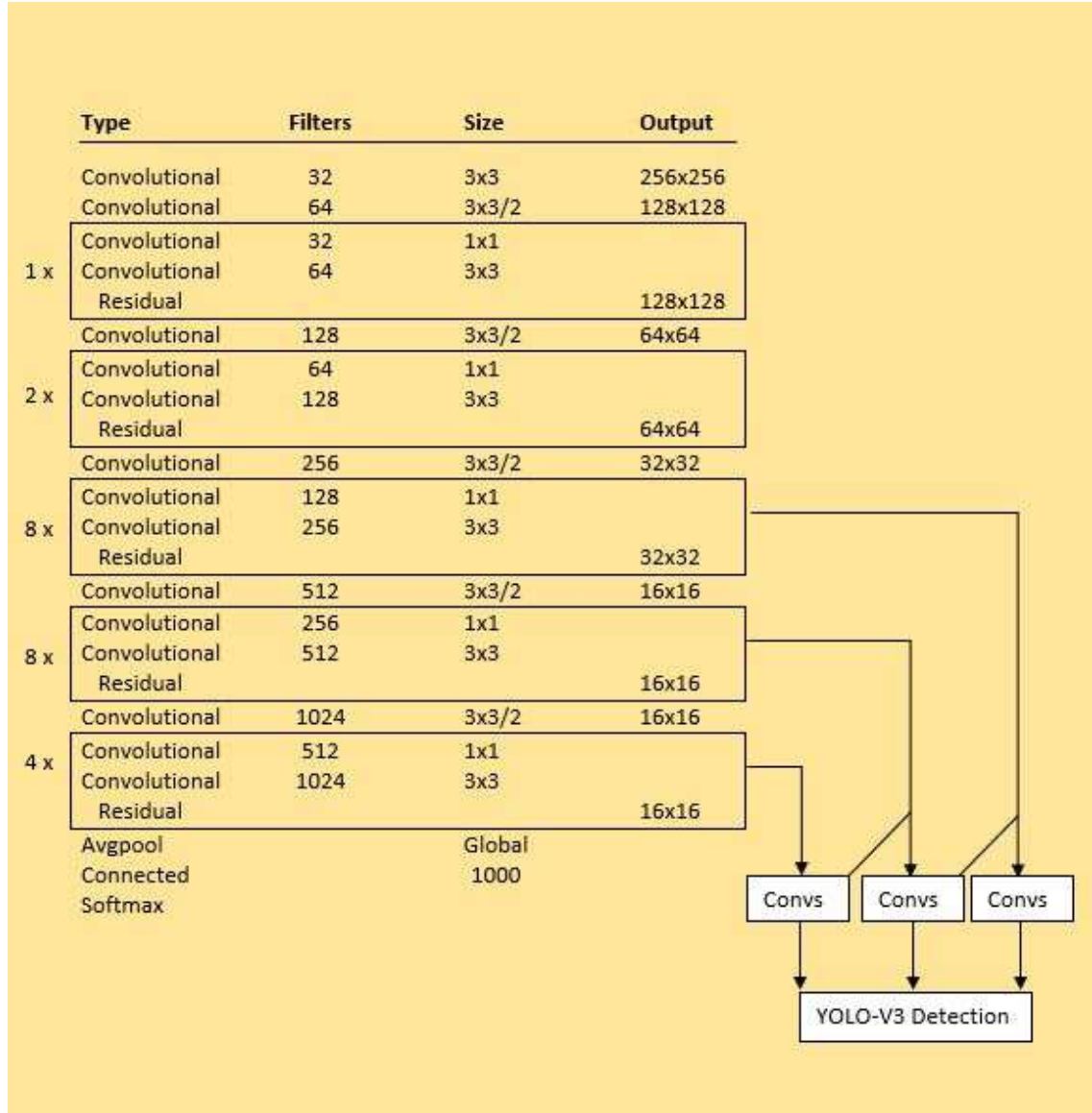
$$\Pr(\text{Class}_i | \text{Object}) * \Pr(\text{Object}) * IOU_{pre}^{truth} = \Pr(\text{Class}_i) * IOU_{pred}^{truth} \quad (2.2)$$

2.2.2. YoloV3

YOLO-V2’de 19 evrişimsel katman ve nesne tespiti yapmak için eklenmiş ekstra 11 katmanı bulunan darknet-19 olarak adlandırılan derin ağ mimarisi kullanılmaktadır. 30 katmanlı bir mimariye sahip YOLO-V2’de genellikle küçük nesneleri algılamada problemler yaşamaktadır. Bunun nedeni girdilerin alt örnekleme katmanlarında özniteliklerini kaybederek tespit aşamasında küçük nesnelere ait özniteliklerin işlenememesidir. YOLO-V2’de düşük seviyeli öznitelikleri yakalamak için ilk katmanlarda bulunan öznitelik haritalarını düşük seviyeli öznitelik haritaları ile birleştiren kimlik eşleme yöntemi kullanılmıştır. Fakat hala YOLO-V2’de artık bloklar(residual blocks), bağlantıyı atla(skip connection) ve yukarı örnekleme(upsampling) gibi özellikler bulunmamaktadır. YOLO-V3 tüm bunları içermektedir.

YOLO-V3 ağı YOLO ve YOLO-V2 ağları kullanılarak geliştirilmiştir. YOLO-V3’de YOLO-V2 gibi Darknet’in türevlerinden biri olan darknet-53 mimarisi kullanılmaktadır. Darknet-53 ImageNet üzerinde eğitilmiş 53 katmandan oluşmaktadır. Tespit etme işlemini gerçekleştirmek için ekstra 53 katman daha kullanarak toplam 106 adet tamamen evrişimsel katman ile YOLO-V3 mimarisi oluşturulmuştur. Bu nedenle YOLO-V2’ye göre çok daha yavaş çalışmaktadır. Özellik çıkarma işlemini gerçekleştirmek için ardışık 3x3 ve 1x1 evrişimsel katmanları ve Artık Ağlar(Residual Networks) [56] fikrinden yararlanılmaktadır. YOLO-V3’te 5 adet artık blok bulunmaktadır. Her bir artık blok

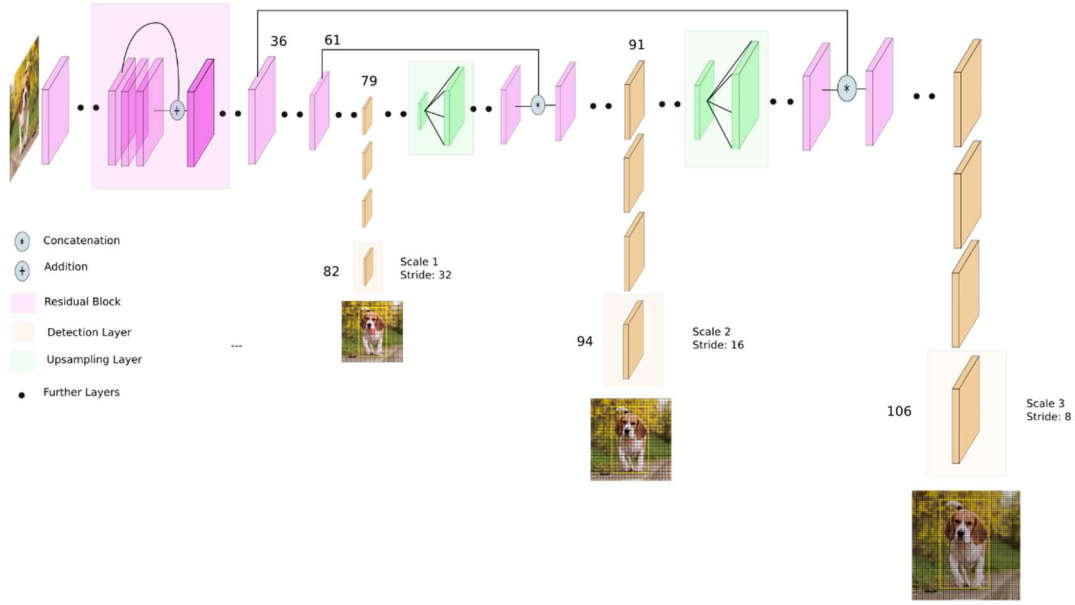
birden fazla artık birimden oluşmaktadır. Şekil 2.11’de Darknet-53 mimarisini görmek mümkündür.



Şekil 2.11. Darknet-53 mimarisi.[23]

YOLO-V3’ün en belirgin özelliği, üç farklı ölçekte tespit yapmasıdır. YOLO tamamen evrimsel bir ağıdır. Tespit işleminin sonucu olarak verilen çıktı, öznetelik haritası üzerine uygulanmış 1x1 algılama çekirdeğinin (detection kernels) sonucuna göre belirlenmektedir. Şekil 2.11’de görüldüğü gibi YOLO-V3’te bu işlem üç farklı yerde üç

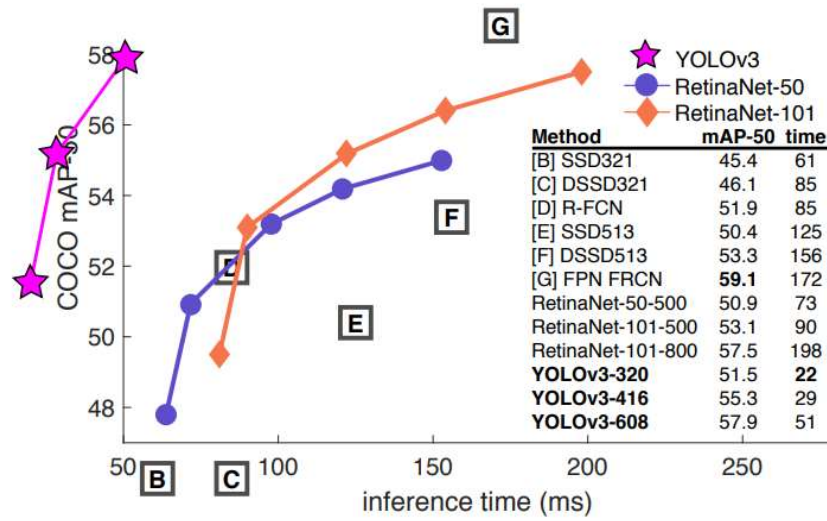
farklı ölçekteki görüntüye 1x1 algılama çekirdeği uygulanarak elde edilir. Algılama çekirdeğinin yapısı $1 \times 1 \times (B \times (5 + C))$ tir. B bir hücre içinde özellik haritası tarafından tahmin edilmiş sınırlayıcı kutuların sayısını, '5' bir nesne güven değeri ve dört sınırlayıcı kutu parametresini ve C sistemde bulunan sınıf sayısını temsil etmektedir. Şekil 2.12'de [57] YOLO-V3 ağ yapısını görmek mümkündür.



Şekil 2.12. YOLO-V3 Ağ Yapısı.[57]

Verilen görüntü beş kez alt örnekleme(downsampling) yapılmaktadır. Şekil 2.12'de görüldüğü gibi son üç alt örnekleme katmanlarında farklı ölçeklerde tespit işlemi yapılmaktadır. İlk tespit işlemi 82. Katmanda yapılmaktadır. Görüntüye ilk 81 katmanda alt örnekleme işlemi yapılmaktadır. 82. Katmanda 32'lik bir adım kaydırma(stride) işlemi yapılarak büyük nesnelerin tespiti yapılmaktadır. Ardından 79. Katmanda bulunan öznetelik haritasına yukarı örnekleme (upsampling) işlemi uygulanarak 61. Katmanda bulunan öznetelik haritası ile birbirine bağlanmaktadır. İlk tespit işleminde yapılan uygulamalar 94.ve ardından 106. Katmanlarda farklı adım kaydırma boyutları ile uygulanarak sırası ile orta boyuttaki ve küçük boyuttaki nesnelerin tespiti yapılmaktadır. Küçük boyuttaki öznetelik haritalı derin anlamsal bilgilere ve büyük boyuttaki öznetelik haritaları hedeflerin daha ince taneli (fine grained) bilgilerine sahip olduğundan öznetelik

kaynaştırması yapılmaktadır. YOLO-V3, yukarı örnekleme(upsampling) yaparak daha derinde kullanılan katmanların öznetelik haritalarını yeniden boyutlandırarak öznetelik kaynaştırmasını gerçekleştirmektedir. YOLO-V3’de farklı ölçekteki öznetelik haritaları aynı boyuta getirildikten sonra, bağlama(concatenation) yöntemi sayesinde derin katmandaki öznetelikler ile ilk katmanlarda elde edilen öznetelikleri birleştirmektedir. Bu sayede YOLO-V3 hem büyük hedeflerin hem de küçük hedeflerin tespit edilmesinde iyi bir performans sağlamaktadır [58]. Şekil 2.13’de YOLO-V3’ün alanında iyi olan diğer nesne tespit etme algoritmaları ile COCO veri seti üzerinde yapılmış olan performans karşılaştırmasını görmek mümkündür.

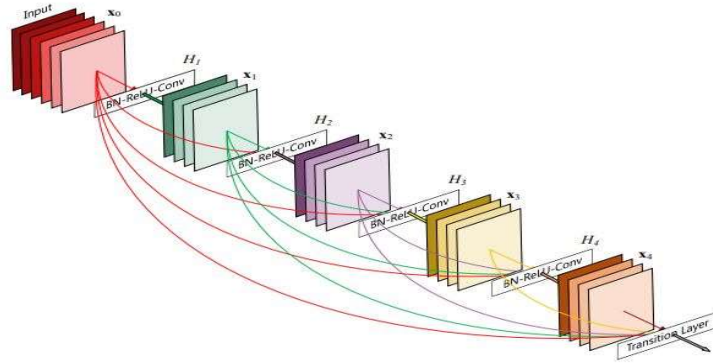


Şekil 2.13. COCO veri seti üzerinde 0.5 IoU değeri ile nesne tespit yöntemlerine ait mAP değerlerinin ve tespit sürelerinin karşılaştırılması.[58]

2.2.3. Yoğun Bağlantılı Sinir Ağları (DenseNet)

Sinir ağları eğitilirken konvolüsyon ve alt-örnekleme işlemleri nedeniyle özellik haritalarında azalma meydana gelmektedir. Aynı zamanda katmanlar arası geçişlerde görüntü özelliğinde kayıplar olmaktadır. Görüntü özellik bilgilerinin daha etkili kullanılması için Huang ve arkadaşları [59] DenseNet sistemini geliştirmişlerdir. Yapılan sistemde her bir katman diğer katmanlara ileri beslemeli bir şekilde bağlanmıştır. Bu sayede herhangi bir l katmanı kendinden önceki bütün katmanlara ait özellik bilgisine

erişebilmektedir. Yoğun bağlantılı sinir ağları genel ağ mimarisini Şekil 2.14’de [59] görmek mümkündür.



Şekil 2.14. Yoğun Bağlantılı Sinir Ağları.[59]

$$X_i = H_i[X_0, X_1, X_2, \dots, X_{i-1}] \quad (2.3)$$

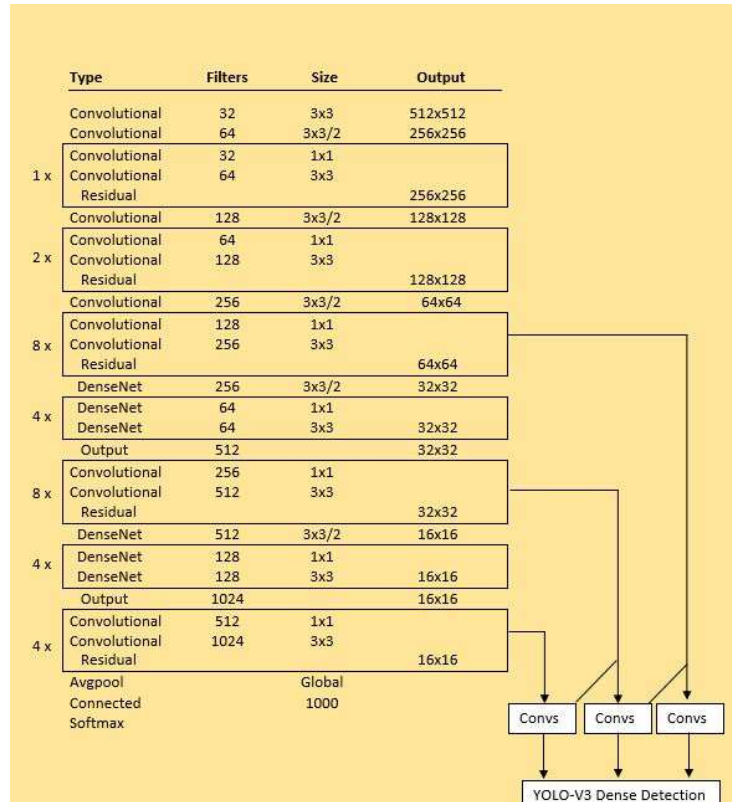
$[X_1, X_2, X_3, \dots, X_{i-1}]$ katmanları, özellik bilgilerinin birleşimidir. H_i , özellik bilgilerini işlemek için kullanılan transfer fonksiyonudur. Bu sayede özelliklerin ağı yayılım oranı artmakta, tekrar kullanımı kolaylaşmakta ve parametre sayısında büyük ölçüde azalma olmaktadır.

2.2.4. Önerilen Algoritma YOLO-V3 Dense

Yapılan çalışmada kullanılan YOLOV3-Dense ağ parametreleri Şekil.8.’de gösterildiği gibidir. Yüksek çözünürlüklü görüntüleri daha iyi işlemek için öncelikle görüntüler 512x512 olarak yeniden boyutlandırılmıştır. İlerleyen katmanlarda görüntüler 32x32 ve 16x16 alt örnekleme filtrelerinden geçirilerek özneteliklerin çıkarılması sağlanmıştır.

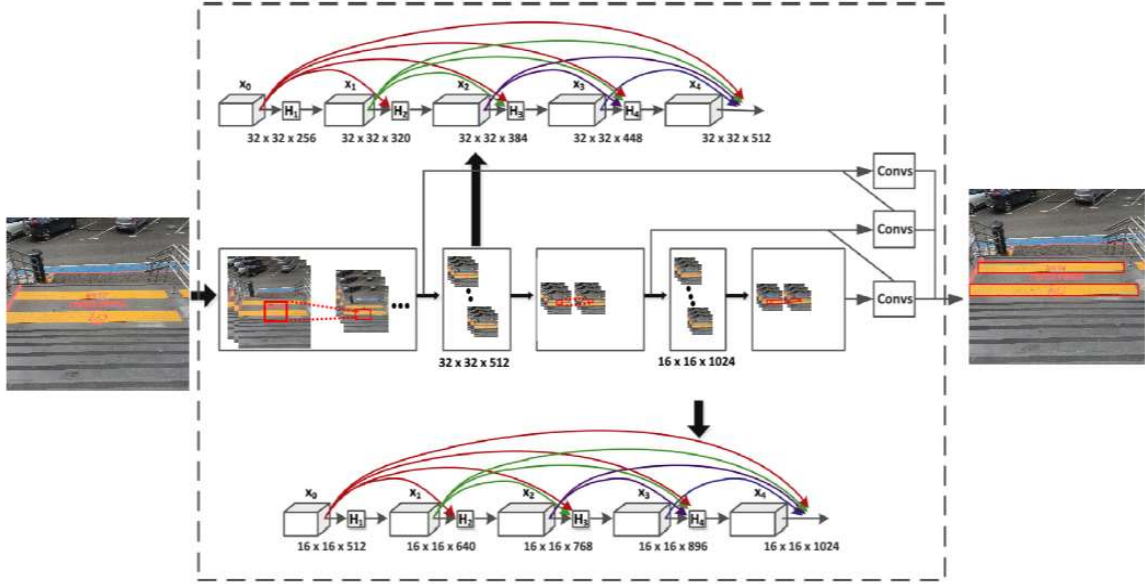
Yapılan çalışmada öznetelik bilgilerini işlemek için kullanılan transfer fonksiyonu H_i , BN-LReLU-Conv(1X1) fonksiyonunu kullanmaktadır. BN (Batch Normalization) yığın normalizasyonu, LReLU(Leaky Rectified Linear Units) ve Conv, konvolüsyon özelliklerinin birleşimidir. H_i doğrusal olmayan $[X_0, X_1, X_2, \dots, X_{i-1}]$ katmanlarını

üretmektedir. Bu katmanların her biri 32x32 çözünürlük değerine sahip 64 özellik katmanından oluşmaktadır. H_1 x_0 üzerinde BN-LReLU-Conv(1X1) fonksiyonunu uygulamaktadır. Ardından çıkan sonuç üzerinde BN-LReLU-Conv(3X3) fonksiyonunu uygulayarak görüntünün öznitelik haritasını çıkarmaktadır. Yukarıda H_1 tarafından yapılan işlemler, H_2 tarafından x_0 ve x_1 üzerinde uygulanarak x_2 sonucu elde edilir. Elde edilen $[x_0, x_1, x_2]$ sonuçları H_3 fonksiyonuna girdi olarak verilerek $[x_0, x_1, x_2, x_3]$ sonucu elde edilir. Yapılan bu işlem Şekil 2.15’de görüldüğü gibi 16x16x1024 öznitelik katmanına kadar tekrarlayarak ilerlemektedir. Bu sayede görüntü öznitelikleri bir katmandan diğerine aktarılarak düşük çözünürlüğe sahip görüntüler üzerinde çalışan katmanlarda da görüntünün bozulmadan önce alınan öznitelikleri kullanılmıştır. Öznitelik kaybı azaltılmış, öznitelik kullanım oranı arttırılmış ve daha iyi bir sonuç alınması sağlanmıştır.



Şekil 2.15. YOLOV3-Dense Ağ parametreleri.

Şekil 2.16’da YOLOV3-Dense mimarisinde Darknet-53 mimarisinin temel ağ mimarisi olarak nasıl kullanıldığı gösterilmektedir. Darknet-53’te bulunan düşük çözünürlüğe sahip orijinal transfer katmanları yerine, öznetelik yayılımını artırmak ve özneteliklerin yeniden kullanılmasını ve özellik haritalarının birleştirmek adına DenseNet kullanılmıştır.



Şekil 2.16. YOLOV3-Dense YSA mimarisi.

2.2.4.1. Kullanılan Teknolojiler

Çalışmada kullanılan veri seti tarafımızca elde edilen görüntülerden oluşmaktadır. Bu görüntüler elde edilirken GoPro Hero 7 aksiyon kamerası ve Sony HDR-X3000R-4K aksiyon kamerası kullanılmıştır. Elde edilen verileri çoğullamak ve etiketlemek için python programlama dili kullanılmıştır. Sistemi ekran kartı üzerinde çalıştırmak için Nvidia tarafından oluşturulmuş ve açık kaynak olarak kullanılabilen, ekran kartı üzerinde paralel hesaplama imkânı sunan CUDA kütüphanesinin 9. Versiyonu kullanılmıştır. Görüntüler üzerinde yapılan işlemlerde açık kaynak olarak kullanılabilen OpenCv kütüphanesi kullanılmıştır.

3. BULGULAR VE TARTIŞMA

Çalışmada tasarlanan yapay sinir ağı modelinin eğitim ve test süreçleri NVIDIA GeForce GTX1080 Ti 11 GB ekran kartı üzerinde gerçekleştirilmiştir. Sistemin düzgün bir şekilde çalışması için gerekli olan momentum, başlangıç öğrenme oranı(initial learning rate), kilo kaybı düzenleme (weight decay regularization) ve diğer parametreler orijinal YOLO-V3 modelindeki ile aynı alınmıştır. Model yaklaşık olarak 160 saat eğitilmiştir. İlk 40000 adımdan sonra öğrenme oranı 0.0001'e ardından 50000 adımdan sonra 0.00001 e düşürülmüştür. Eğitim işlemi 70000 dönem(epoch) sürmüş ve her 10000 adımda modelin ağırlıkları yedeklenmiştir. Yedeklenen bu ağırlıklar daha sonra test verileri üzerinde çalıştırılarak ağırlığa ait ortalama hassasiyet (average precision) değerlerinin ortalamaları(mean Average Precision) karşılaştırılmıştır. Eğitilen modelin performansını değerlendirmek için sistemin kesinlik(precision), duyarlılık(recall), yitim(loss), F1 skoru, ortalama hassasiyet (mAP) ve intersection over union (IoU) değerlerine bakılmıştır.

3.1. Kullanılan Metrik Yöntemler

3.1.1. Kesinlik (Precision), Duyarlılık (Recall) , F1 skor ve mAP

Olasılık modelini kullanan tüm sistemler, sisteme ait örnekleri dört ayrı başlık altında toplamaktadır [60]. Bunlar doğru pozitif(TP), yanlış pozitif(FP), doğru negatif(TN) ve yanlış negatif(FN) değerleridir. Bu değerler, kullanıcı tarafından etiketlenmiş verinin etiket değeri ve eğitilmiş sistem tarafından yapılan tahmin sonucunun uyuşup uyuşmadığına göre oluşmaktadır. Etiketlenmiş verinin pozitif olduğu varsayılırsa, tahmin sonucu eğer pozitif ise TP, negatif ise FN değeri oluşmaktadır. Yine aynı şekilde etiketlenmiş verinin negatif olduğu varsayılırsa, tahmin sonucu eğer pozitif ise FP, negatif ise TN değeri oluşmaktadır. Karmaşıklık matrisini tablo.1'de görmek mümkündür.

Tablo 3.1. Karmaşıklık Matrisi.

	Etiketlenmiş Durum		
	Toplam Durum	Etiket Pozitif	Etiket Negatif
Tahmin Edilen Durum	Tahmin Pozitif	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
	Tahmin Negatif	Yanlış Negatif (FN)	Doğru Negatif (TN)

Kesinlik (Precision): Tahmin edilen örneklerdeki doğru pozitif sonuçların, pozitif olarak tahmin edilen tüm sonuçlara oranı ile hesaplanır. Kesinlik değerinin formülünü Denklem 3.1’de görülmektedir.

$$P = \frac{DP}{DP+YP} \quad (3.1)$$

Duyarlılık (Recall): Tahmin edilen örneklerdeki doğru pozitif sonuçların, olması gereken tüm pozitif sonuçlara oranı ile hesaplanır. Duyarlılık değerinin formülünü Denklem 3.2’de görülmektedir.

$$R = \frac{DP}{DP+YN} \quad (3.2.)$$

F1-skoru: Yukarıda verilen değerlerin harmonik ortalaması olarak hesaplanmaktadır. F-1 skorunun formülünü Denklem 3.3’de görülmektedir.

$$F_1Skoru = 2x \frac{PxR}{P+R} \quad (3.3)$$

Ortalama Averaj Kesinliği (mAP): Kesinlik, duyarlılık, F1-skoru gibi değerlerin tek bir noktadan değerlendirilmesi için tasarlanmış metrik bir sistemdir. mAP değerinin formülünün Denklem 3.4’de görülmektedir.

$$mAP = \int_0^1 P(R)dR \quad (3.4)$$

3.1.2. Kayıp Fonksiyonu (Loss Function)

YOLO modelinde her bir hücre(grid) birden fazla sınırlayıcı kutu üretmektedir. Doğru pozitif olan durumlarda yitim değerini hesaplamak için üretilen sınırlayıcı kutular içerisinde en yüksek IoU değerine sahip olan kutu seçilmelidir. Bu sayede gelecek jenerasyonlarda düşük IoU değerine sahip sınırlayıcı kutular elenerek daha yüksek IoU değerine sahip olan sınırlayıcı kutular üretilmektedir. YOLO yitim değerini hesaplamak için, tahmin edilen ve gerçek referans değer olan sınırlayıcı kutular arasındaki sınıf, yerini saptama ve güven değerlerinin kare hatalarının toplamını almaktadır. Bu değerlere; sınıflandırma yitimi, yerini saptama yitimi ve güven yitimi denilmektedir[23].

3.1.2.1. Sınıflandırma Yitimi

Sınıflandırma yitiminde nesne tespit edildiğinde her bir sınıf için koşullu sınıf olasılıklarının kare hatası hesaplanmaktadır. Sınıflandırma yitimi denklemini Denklem 3.5'te görmek mümkündür.

$$\sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in sınıflar} (P_i(C) - \hat{p}_i(C))^2 \quad (3.5)$$

1_i^{obj} : Hücrede nesne varsa 1 yoksa 0 olmaktadır.

$\hat{p}_i(C)$: i hücresindeki c sınıfı için koşullu sınıf olasılığı.

3.1.2.2. Yerini Saptama Yitimi

Yerini saptama yitimi, tahmin edilen sınırlayıcı kutunun konumu ve boyut ile ilgili yitim değerini hesaplanmaktadır. Yalnızca nesneyi tespit eden kutu için hesaplanmaktadır. Yerini saptama yitimi denklemini Denklem 3.6'da görmek mümkündür.

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \quad (3.6)$$

1_{ij}^{obj} : i . hücredeki j . sınırlayıcı kutu nesne algılamaktan sorumlu ise 1 değilse 0.

λ_{coord} : Sınırlayıcı kutu koordinatlarındaki yitim değerinin ağırlığını artırır.

3.1.2.3. Güven Yitimi

Güven yitimi, nesne tespit edildiği zaman tespit eden kutunun güven değerini hesaplamaktadır. Güven yitimi denklemini Denklem 3.7’de görmek mümkündür.

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (3.7)$$

\hat{C}_i : i hücresinde j . kutunun güven değeri .

1_{ij}^{obj} : i . hücredeki j . sınırlayıcı kutu nesne algılamaktan sorumlu ise 1 değilse 0.

Eğer kutuda nesne tespit edilmezse güven yitimi:

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (3.8)$$

\hat{C}_i : i hücresinde j . kutunun güven değeri .

1_{ij}^{noobj} : 1_{ij}^{obj} değerinin tümleyenidir.

λ_{noobj} : Arka plan tespit edildiğinde yitimi azaltmak için kullanılır.

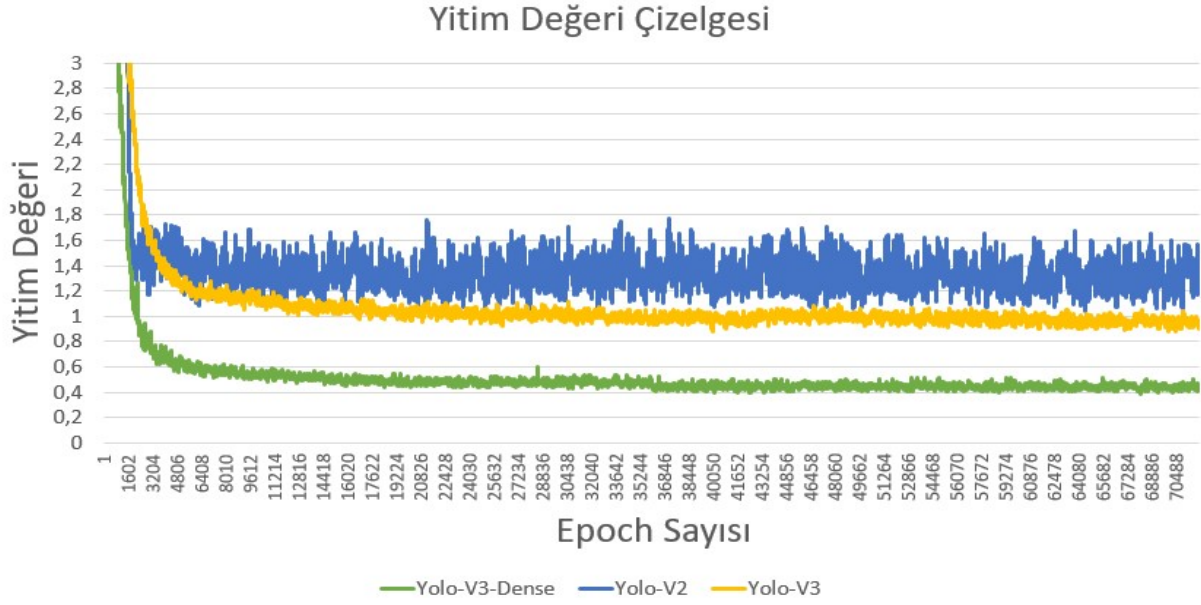
Son yitim değeri; sınıflandırma, yerini saptama ve güven yitimi değerlerinin toplamıdır.

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2 \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{C \in sınıflar} (P_i(C) - \hat{p}_i(C))^2 \end{aligned} \quad (3.9)$$

3.2. Modellerin Karşılaştırılması

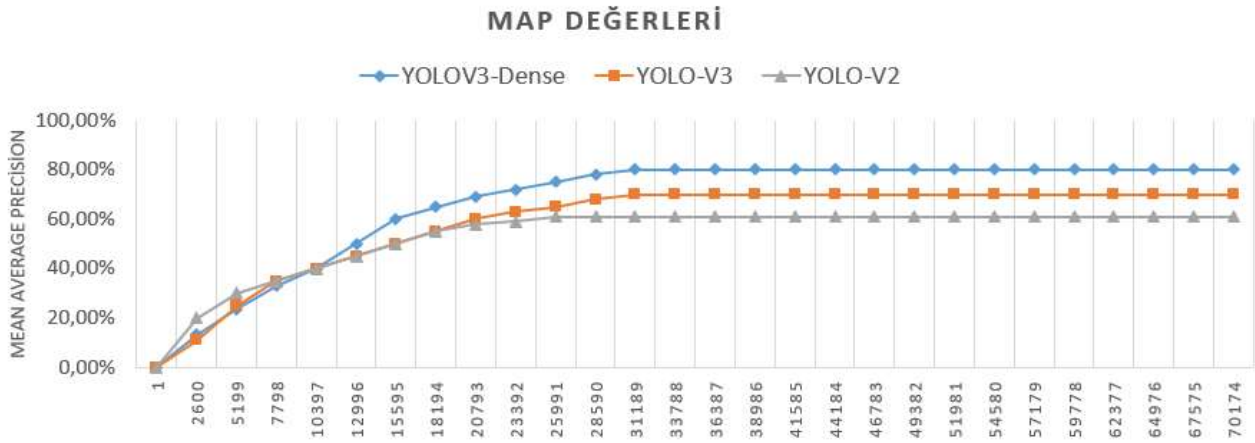
Yapılan çalışmaya ait yitim değerleri çizelgesini Şekil.3.1’de görmek mümkündür. Dokunsal parke yüzeyi tespitinde modellerin performanslarının kıyaslanması için, YOLO-V2, YOLO-V3 ve YOLOV3-Dense modelleri oluşturulan veri seti üzerinde çalıştırılmıştır. Eğitim işlemi sırasında kullanılan yitim değerleri kayıt edilerek aşağıda görülen Şekil.3.1’de Yitim Değerleri Çizelgesinde birbirleri ile kıyaslanmıştır. Son yitim değerleri yaklaşık olarak YOLO-V2 modeli için 1.15 YOLO-V3 modeli için 0,82 ve

YOLOV3-Dense için 0,36'dır. Orijinal YOLO-V3 ve YOLOV3-Dense arasında yaklaşık olarak 0,46 fark bulunmaktadır.



Şekil 3.1. Modellere ait yitim değerleri.

Güven skoru eşik değeri 0,25 olarak alınıp ortalama hassasiyet değerleri karşılaştırıldığında YOLOV3-Dense modelinin %81,2 değeri ile orijinal YOLO-V3 modelinden yaklaşık olarak %10 daha iyi olduğu Şekil.3.2 ortalama hassasiyet değerleri çizelgesinde görülmektedir. YOLO-V2, YOLO-V3, YOLOV3-Dense modellerine ait F1-Skor, IoU, mAP ve Ortalama tespit zamanı değerleri Tablo 3.2’de görülmektedir. Çizelge



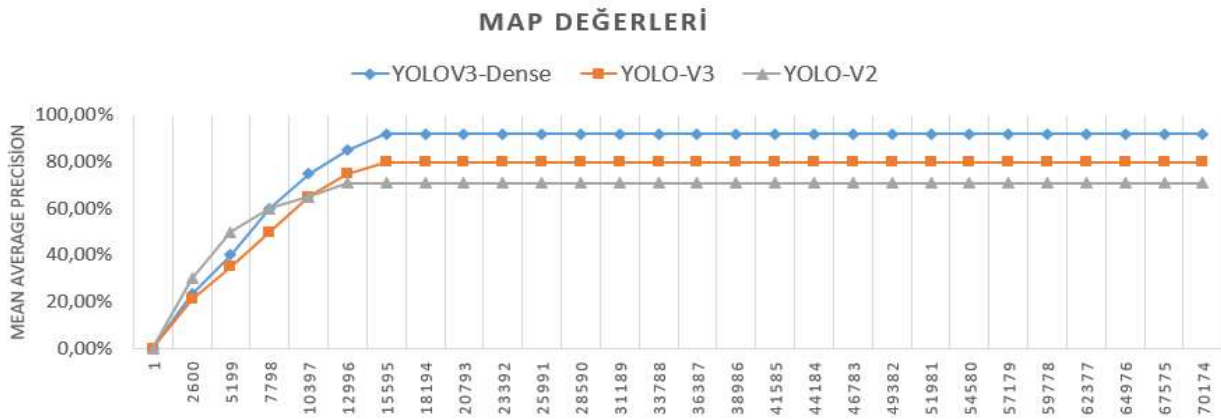
Şekil 3.2. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,25

ve tablolara bakıldığında YOLOV3-Dense modelinin sınırlayıcı kutu tespiti işleminde diğer iki modelden daha iyi olduğu görülmüştür.

Tablo 3.2. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri.

Modeller	F1-Skor	mAP	IoU	Tespit Zamanı(s)	Güven Skoru Eşik Değeri
YOLO-V2	0,701	0,60	0,50	0,201	0,25
YOLO-V3	0,721	0,71	0,61	0,232	0,25
YOLOV3-Dense	0,814	0,80	0,70	0,254	0,25

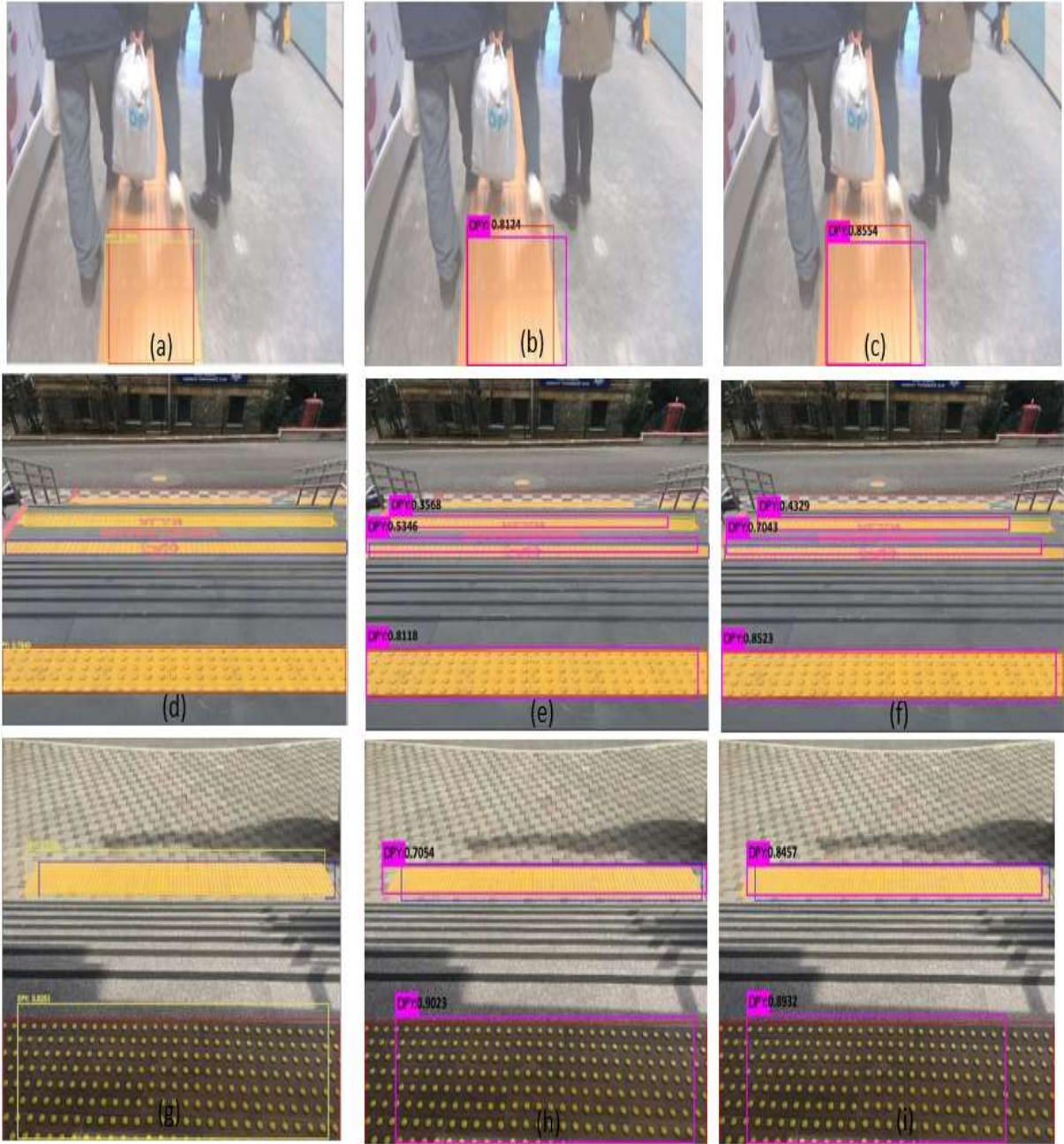
Güven skoru eşik değeri 0,50 olarak belirlenerek bu değer altında kalan sınırlayıcı kutular göz ardı edilmiş ve mAP, F1-skor ve IoU değerleri hesaplanırken bu kutular işleme alınmamıştır. Güven skoru eşik değerinde yapılan bu değişim ile YOLO-V2, YOLO-V3 ve YOLOV3-Dense modellerinin tümünde mAP, F1-skor ve IoU değerlerinin yaklaşık olarak %10 daha iyi performans gösterdiği Tablo 3.3’de görülmektedir.



Şekil 3.3. Ortalama Hassasiyet Değerleri Çizelgesi, Güven Skoru Eşik Değeri=0,50

Tablo 3.3. Modellere ait F1-Skor, mAP, IoU ve tespit zamanı değerleri.

Modeller	F1-Skor	mAP	IoU	Tespit Zamanı(s)	Güven Skoru Eşik Değeri
YOLO-V2	0,69	0,71	0,58	0,201	0,50
YOLO-V3	0,78	0,80	0,68	0,232	0,50
YOLOV3-Dense	0,89	0,92	0,81	0,254	0,50

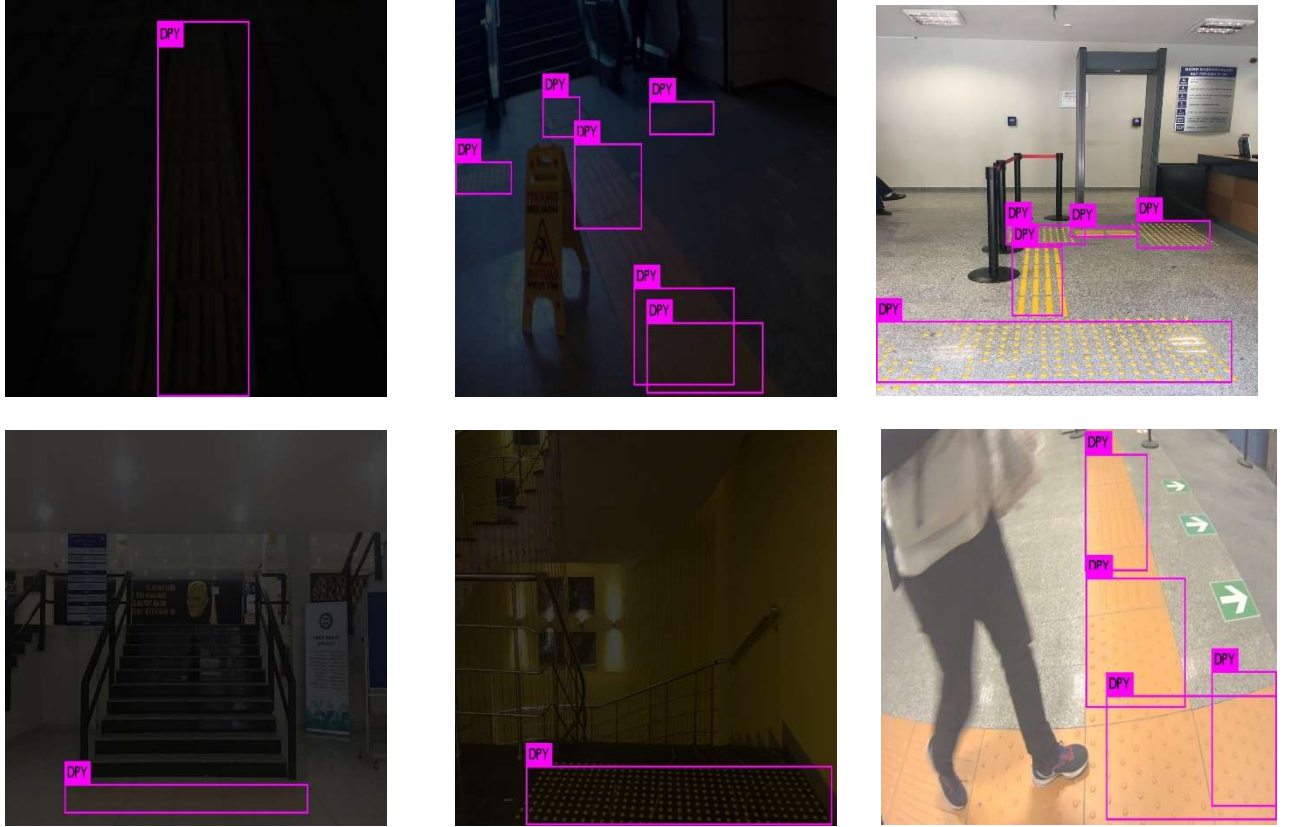


Şekil 3.4. DPY Tespit Edilmiş Örnek Görüntüler.

DPY tespit edilmiş 3 adet örnek görüntü Şekil 3.4’te görülmektedir. Her bir görüntü 3 model ile ayrı ayrı test edilerek çıkan sonuçlar Tablo 3.4’te toplanmıştır. a, d, g görselleri YOLO-V2 ile b, e, h görüntüleri YOLO-V3 ile c, f, i görüntüleri YOLOV3-Dense ile tespit edilmiştir.

Tablo 3.4. Örnek veri setine ait güven skoru değerleri.

Görüntü	Kullanılan Model	Güven Skoru
a	YOLO-V2	0,79
b	YOLO-V3	0,81
c	YOLOV3-Dense	0,85
d	YOLO-V2	0,78-0-0
e	YOLO-V3	0,85-0,53-0,35
f	YOLOV3-Dense	0,85-0,70-0,43
g	YOLO-V2	0,82-0,80
h	YOLO-V3	0,90-0,70
i	YOLOV3-Dense	0,89-0,84



Şekil 3.5. Doğrulama veri setine ait örnek görüntüler.

Tablo 3.5. Diğer yöntemler ile karşılaştırma tablosu.

Çalışma	Kullanılan Yöntem	Doğruluk	fps	Avantajlar	Dezavantajlar
[4]	Canny, Laplace, Sobel Kenar Belirleme Algoritmaları ve Karar Ağaçları	%88,48	16,27	Geleneksel görüntü işleme algoritmalarına ait örnekler mevcuttur.	Düşük fps değeri gerçek zamanlı sistemlerde kullanıma uygun değildir.
[49]	Gri seviyeli eş-oluşum matrisleri ve Destek Vektör Makineleri	-	-	Danışmanlı öğrenme yöntemlerine ait örnekler mevcuttur.	Sistemin çıktıları ile ilgili herhangi bir bilgi verilmemiştir.
[50]	Çok Renkli Görüntü Segmentasyonu ve Kirsch Kenar Belirleme Algoritması.	-	2	Geleneksel görüntü işleme algoritmaları ve donanımsal parçaların bir arada kullanılmasına ait örnekler mevcuttur.	Düşük fps değerine sahiptir. Günlük hayatta kullanımı oldukça masraflı ve zor bir sistemdir.
[24]	YOLO-V2	%71(mAP)	70	Diğer YOLO modellerine kıyasla daha sıkı bir ağ yapısı olduğundan fps değeri yüksektir.	Doğruluk ve mAP değerleri düşüktür.
[22]	YOLO-V3	%80(mAP)	60	Çok katmanlı ağ yapılarına ait örnekler mevcuttur.	Görüntü üzerindeki küçük nesnelerin tespit edilmesinde doğruluk değeri düşüktür.
	YOLOV3-Dense	%92(mAP)	60	DenseNet sayesinde görüntüde bulunan tüm öznitelikler ağda bulunan tüm katmanlarda tekrar tekrar kullanılmış ve görüntü üzerindeki küçük nesnelerin tespiti sağlanmıştır.	Görüntü üzerindeki tüm özniteliklerin kullanılması ağda bulunan parametre sayısını artırmıştır.

Tablo 3.5’te görüldüğü gibi YOLOV3-Dense modeli, geleneksel görüntü işleme yöntemleri ile yapılan sistemlere göre daha iyi sonuç vermektedir. Geleneksel yöntemler il yapılan sistemlerin en büyük dezavantajının düşük fps değeri olduğu görülmektedir. Bu dezavantaj nedeni ile yapılan sistemler gerçek zamanlı olarak kullanılamamaktadır. YOLOV3-Dense modelinde yalnızca doğruluk değeri büyük değil aynı zamanda fps değeride geleneksel yöntemlere göre oldukça yüksektir. Bu sayede gerçek zamanlı çalışan sistemlerde kullanılabilmektedir.

4. SONUÇLAR

Bu çalışmada, nesne tespit etme alanında kullanılan YOLO-V3 modeli DenseNet modeli ile birleştirilerek dokunsal parke yüzeylerinin tespiti yapılmıştır. Bu model ile dokunsal parke yüzeylerinin tespit edilmesi ve görüntü içerisinde hangi konumda tespit edilebilmektedir. YOLOV3-Dense modeli ile YOLO-V3 modelinde bulunan öznelik katmanlarındaki düşük çözünürlüğe sahip görüntülerin iyileştirilmesi amaçlanmıştır. Bu amaç doğrultusunda görüntüde bulunan özneliklerin tüm katmanlarda eşit bir şekilde yayılması DenseNet yöntemi ile sağlanmış ve ağ performansı artırılmıştır. Deneysel çalışmalar sonucunda YOLOV3-Dense modelinin %89 F1-skor, %92 ortalama hassasiyet ve %81 IoU değerleri ile diğer YOLO modellerinden çok daha iyi sonuç verdiği anlaşılmaktadır.

Gelecek çalışmalarda tarafımızca oluşturulmuş olan Marmara Dokunsal Parke Yüzeyi (MDPY) veri seti genişletilerek daha iyi sonuçlar alınması planlanmaktadır. Bu çalışmada kullanılan veri artırma yöntemleri ve tespit etme algoritmaları optimize edilerek tespit etme doğruluğunun artırılması ve çalışma sonucunda elde edilen evrimsel sinir ağı ağırlıklarının farklı platformlarda kullanılarak gündelik hayatta kullanılabilen mobil uygulamalar geliştirilmesi planlanmaktadır. Gerçek zamanlı olarak tespit edilen dokunsal parke yüzeyleri konumunun sistem tarafından seslendirilmesi ile engelli bireylere yönlendirme yapılarak etkili bir yardımcı teknoloji olarak kullanılması sağlanabilir. Gelecek çalışmalarda MDPY veri setinde insanlar, araçlar, ağaçlar, tabelalar gibi birçok nesne etiketlenerek, görme engelli bireylere etrafta bulunan engeller ve nesneler hakkında bilgilendirme yapılabilir. Yapılması planlanan uygulamalar sayesinde kullanıcı ekstra herhangi bir donanımına ihtiyaç duymadan yalnızca akıllı telefon ile sistemi kullanabilir. Yapılması planlanan uygulamalar ile yanlış kurulum yapılan DPY'lerinin konumları GPS yardımı ile belirlenerek ilgili belediye, kurum ve yetkili kişilere iletilerek oluşan mağduriyet giderilebilir. DPY yollarının kullanılmasını engelleyen araç, kişi ve nesneler otonom bir şekilde algılanarak yetkili kişiler bilgilendirilebilir ve kullanımı engelleyen kişiler hakkında cezai işlemler uygulanabilir.

KAYNAKLAR

- [1] World Health Organization: <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment> Erişim Tarihi: 12.01.2019
- [2] Lu, J., Siu, K. W. M., & Xu, P. (2008). *A comparative study of tactile paving design standards in different countries*. Paper presented at the Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on, Kunming, China.
- [3] Asami, T., & Ohnishi, K. (2014). *Crosswalk location, direction and pedestrian signal state extraction system for assisting the expedition of person with impaired vision*. Paper presented at the Mecatronics (MECATRONICS), 2014 10th France-Japan/8th Europe-Asia Congress on, Tokyo.
- [4] Ghilardi, M. C., Macedo, R. C. O., & Manssour, I. H. (2016). A New Approach for Automatic Detection of Tactile Paving Surfaces in Sidewalks. *Procedia Computer Science*, 80, 662-672. doi: <https://doi.org/10.1016/j.procs.2016.05.356>
- [5] Mancini, A., Frontoni, E., & Zingaretti, P. (2018). Mechatronic System to Help Visually Impaired Users During Walking and Running. *IEEE Transactions on Intelligent Transportation Systems*, 19(2), 649-660. doi: 10.1109/TITS.2017.2780621
- [6] Shoval, S., Borenstein, J., & Koren, Y. (1998). The NavBelt-a computerized travel aid for the blind based on mobile robotics technology. *IEEE Transactions on Biomedical Engineering*, 45(11), 1376-1386. doi: 10.1109/10.725334
- [7] Yang, K., Cheng, R., Wang, K., & Zhao, X. (2015). *A ground and obstacle detection algorithm for the visually impaired*. Paper presented at the 2015 IET International Conference on Biomedical Image and Signal Processing (ICBISP 2015), Beijing, China.
- [8] Tsai, C.-W., Lai, C.-F., Chao, H.-C., & Vasilakos, A. (2015). Big data analytics: A survey. *Journal of Big Data*, 2. doi: 10.1186/s40537-015-0030-3
- [9] Garcia-Garcia, A., Orts, S., Oprea, S., Villena Martinez, V., Martinez-Gonzalez, P., & Rodríguez, J. (2018). A Survey on Deep Learning Techniques for Image and Video Semantic Segmentation. *Applied Soft Computing*, 70, 41-65. doi: 10.1016/j.asoc.2018.05.018
- [10] Krizhevsky, A., Sutskever, I., & E. Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25. doi: 10.1145/3065386
- [11] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Li, F. F., & Ieee (2009, Jun 20-25). *ImageNet: A Large-Scale Hierarchical Image Database*. Paper presented at the IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition Workshops, Miami Beach, FL.

- [12] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Li, F. F. (2014). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115. doi: 10.1007/s11263-015-0816-y
- [13] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*. Paper presented at the Proceedings of the IEEE international conference on computer vision.
- [14] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv preprint arXiv:1905.11946*.
- [15] McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4), 12.
- [16] Öztemel, E. *Yapay Sinir Ağları*.
- [17] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436. doi: 10.1038/nature14539
- [18] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. doi: 10.1038/nature14539
- [19] <http://cs231n.github.io/neural-networks-1/> Erişim Tarihi: 12.01.2019
- [20] Öztemel, E. (2003). *Yapay Sinir Ağları*. Istanbul: Papatya Yayıncılık.
- [21] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. [Review]. *Proceedings of the Ieee*, 86(11), 2278-2324. doi: 10.1109/5.726791
- [22] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [23] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- [24] Redmon, J., & Farhadi, A. (2017, 21-26 July 2017). *YOLO9000: Better, Faster, Stronger*. Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [25] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- [26] Mizuno, T., Nishidate, A., Tokuda, K., & Arai, K. (2008). INSTALLATION ERRORS AND CORRECTIONS IN TACTILE GROUND SURFACE INDICATORS IN EUROPE, AMERICA, OCEANIA AND ASIA. *IATSS Research*, 32(2), 68-80. doi: [https://doi.org/10.1016/S0386-1112\(14\)60210-7](https://doi.org/10.1016/S0386-1112(14)60210-7)

- [27] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). *Ssd: Single shot multibox detector*. Paper presented at the European conference on computer vision.
- [28] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.
- [29] Xiao, J., Liu, Z., Zhao, P. F., Li, Y., & Huo, J. W. (2018). Deep Learning Image Reconstruction Simulation for Electromagnetic Tomography. [Article]. *Ieee Sensors Journal*, 18(8), 3290-3298. doi: 10.1109/jsen.2018.2809485
- [30] Pasquet-Itam, J., & Pasquet, J. (2018). Deep learning approach for classifying, detecting and predicting photometric redshifts of quasars in the Sloan Digital Sky Survey stripe 82. [Article]. *Astronomy & Astrophysics*, 611, 11. doi: 10.1051/0004-6361/201731106
- [31] Hagita, K., Higuchi, T., & Jinnai, H. (2018). Super-resolution for asymmetric resolution of FIB-SEM 3D imaging using AI with deep learning. [Article]. *Scientific Reports*, 8, 8. doi: 10.1038/s41598-018-24330-1
- [32] Fang, R. G., Pouyanfar, S., Yang, Y. M., Chen, S. C., & Iyengar, S. S. (2016). Computational Health Informatics in the Big Data Age: A Survey. [Article]. *Acm Computing Surveys*, 49(1), 36. doi: 10.1145/2932707
- [33] Noor, A. M., Holmberg, L., Gillett, C., & Grigoriadis, A. (2015). Big Data: the challenge for small research groups in the era of cancer genomics. [Review]. *British Journal of Cancer*, 113(10), 1405-1412. doi: 10.1038/bjc.2015.341
- [34] Andreu-Perez, J., Poon, C. C. Y., Merrifield, R. D., Wong, S. T. C., & Yang, G. Z. (2015). Big Data for Health. [Article]. *Ieee Journal of Biomedical and Health Informatics*, 19(4), 1193-1208. doi: 10.1109/jbhi.2015.2450362
- [35] Pang, S. C., Orgun, M. A., & Yu, Z. Z. (2018). A novel biomedical image indexing and retrieval system via deep preference learning. [Article]. *Computer Methods and Programs in Biomedicine*, 158, 53-69. doi: 10.1016/j.cmpb.2018.02.003
- [36] Sodemann, A. A., Ross, M. P., & Borghetti, B. J. (2012). A Review of Anomaly Detection in Automated Surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1257-1272. doi: 10.1109/TSMCC.2012.2215319
- [37] Bertini, M., Del Bimbo, A., & Seidenari, L. (2012). Multi-scale and real-time non-parametric approach for anomaly detection and localization. *Computer Vision and Image Understanding*, 116(3), 320-329. doi: <https://doi.org/10.1016/j.cviu.2011.09.009>
- [38] Xu, D., Ricci, E., Yan, Y., Song, J., & Sebe, N. (2015). *Learning Deep Representations of Appearance and Motion for Anomalous Event Detection*.

- [39] Ribeiro, M., Lazzaretti, A. E., & Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. [Article]. *Pattern Recognition Letters*, 105, 13-22. doi: 10.1016/j.patrec.2017.07.016
- [40] Krizhevsky, A., Sutskever, I., & E. Hinton, G. (2012). *ImageNet Classification with Deep Convolutional Neural Networks* (Vol. 25).
- [41] Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015, 7-12 June 2015). *Going deeper with convolutions*. Paper presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [42] Perlin, H. A., & Lopes, H. S. (2015). Extracting human attributes using a convolutional neural network approach. *Pattern Recognition Letters*, 68, 250-259. doi: <https://doi.org/10.1016/j.patrec.2015.07.012>
- [43] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A. K., & Davis, L. S. (2016, 27-30 June 2016). *Learning Temporal Regularity in Video Sequences*. Paper presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [44] Mahadevan, V., Li, W., Bhalodia, V., & Vasconcelos, N. (2010, 13-18 June 2010). *Anomaly detection in crowded scenes*. Paper presented at the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- [45] Cong, Y., Yuan, J., & Liu, J. (2011, 20-25 June 2011). *Sparse reconstruction cost for abnormal event detection*. Paper presented at the CVPR 2011.
- [46] Yu, H., & Winkler, S. (2013, 3-5 July 2013). *Image complexity and spatial information*. Paper presented at the 2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX).
- [47] Shen, J., Liu, N., Sun, H., Tao, X., & Li, Q. (2019). Vehicle Detection in Aerial Images Based on Hyper Feature Map in Deep Convolutional Network. *KSI Transactions on Internet & Information Systems*, 13(4).
- [48] Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster r-cnn: Towards real-time object detection with region proposal networks*. Paper presented at the Advances in neural information processing systems.
- [49] Einloft, D. C., Ghilardi, M. C., & Manssour, I. H. (2016). *Automatic Detection of Tactile Paving Surfaces in Indoor Environments*. Paper presented at the Workshop of Undergraduate Works (WUW) in the 29th Conference on Graphics, Patterns and Images (SIBGRAPI'16).
- [50] Jie, X., Xiaochi, W., & Zhigang, F. (2010). *Research and implementation of blind sidewalk detection in portable eta system*. Paper presented at the Information Technology and Applications (IFITA), 2010 International Forum on.

- [51] Kassim, A. M., Yasuno, T., Mohd Aras, M. S., hj shukor, A., Jaafar, H. I., Baharom, F., & Jafar, F. (2015). *Vision based of tactile paving detection method in navigation system for blind person* (Vol. 77).
- [52] Lv, J.-J., Shao, X.-H., Huang, J.-S., Zhou, X.-D., & Zhou, X. (2017). Data augmentation for face recognition. *Neurocomputing*, 230, 184-196. doi: <https://doi.org/10.1016/j.neucom.2016.12.025>
- [53] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. doi: 10.1186/s40537-019-0197-0
- [54] <https://github.com/puzzledqs/BBox-Label-Tool> Erişim Tarihi: 12.01.2019
- [55] Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.
- [56] He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- [57] <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b> Erişim Tarihi: 12.01.2019
- [58] Ju, Yun, L., Wang, J., Hui, & Chang (2019). The Application of Improved YOLO V3 in Multi-Scale Target Detection. *Applied Sciences*, 9, 3775. doi: 10.3390/app9183775
- [59] Huang, G., Liu, Z., Maaten, L. v. d., & Weinberger, K. Q. (2017, 21-26 July 2017). *Densely Connected Convolutional Networks*. Paper presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [60] Goutte, C., & Gaussier, E. (2005). *A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation*. Paper presented at the Proceedings of the 27th European conference on Advances in Information Retrieval Research.

ÖZGEÇMİŞ

Adı Soyadı : Abdulsamet AKTAŞ
Doğum Yeri : Erzurum
Yabancı Dili : İngilizce
E-Posta : abdulsamet.aktas@marmara.edu.tr

Öğrenim Durumu

Derece	Bölüm/Program	Üniversite/Lise	Mezuniyet Yılı
Lise	Fen Bilimleri	Erzurum Anadolu Lisesi	2009
Lisans	Bilgisayar Mühendisliği	Fırat Üniversitesi	2014

İş Deneyimi

Yıl	Firm/Kurum	Görevi
2014	ACCRIŞ BİLİŞİM YAZILIM DANIŞMANLIĞI	Software Developer
2017	Marmara Üniversitesi	Araştırma Görevlisi